# [MS-DTSX2-Diff]:

# Data Transformation Services Package XML Version 2 File Format

**Intellectual Property Rights Notice for Open Specifications Documentation**

- **Technical Documentation.** Microsoft publishes Open Specifications documentation ("this documentation") for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.
- **Copyrights**. This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.
- **No Trade Secrets**. Microsoft does not claim any trade secret rights in this documentation.
- **Patents**. Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft [Open Specifications Promise](#) or the [Microsoft Community Promise](#). If you would prefer a written license, or if the technologies described in this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting [iplg@microsoft.com](mailto:iplg@microsoft.com).
- **License Programs**. To see all of the protocols in scope under a specific license program and the associated patents, visit the [Patent Map](#).
- **Trademarks**. The names of companies and products contained in this documentation might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit [www.microsoft.com/trademarks](http://www.microsoft.com/trademarks).
- **Fictitious Names**. The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

**Reservation of Rights**. All other rights are reserved, and this notice does not grant any rights other than as specifically described above, whether by implication, estoppel, or otherwise.

**Tools**. The Open Specifications documentation does not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments, you are free to take advantage of them. Certain Open Specifications documents are intended for use in conjunction with publicly available standards specifications and network programming art and, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

**Support.** For questions and support, please contact [dochelp@microsoft.com](mailto:dochelp@microsoft.com).

## Revision Summary

| Date | Revision History | Revision Class | Comments |
|------|------------------|----------------|----------|
| 2/9/2011 | 0.1 | New | Released new document. |
| 7/7/2011 | 0.1 | Major | Significantly changed the technical content. |
| 11/3/2011 | 0.1 | None | No changes to the meaning, language, or formatting of the technical content. |
| 1/19/2012 | 0.1 | None | No changes to the meaning, language, or formatting of the technical content. |
| 2/23/2012 | 0.1 | None | No changes to the meaning, language, or formatting of the technical content. |
| 3/27/2012 | 0.1 | None | No changes to the meaning, language, or formatting of the technical content. |
| 5/24/2012 | 0.1 | None | No changes to the meaning, language, or formatting of the technical content. |
| 6/29/2012 | 0.1 | None | No changes to the meaning, language, or formatting of the technical content. |
| 7/16/2012 | 0.1 | None | No changes to the meaning, language, or formatting of the technical content. |
| 10/8/2012 | 0.1 | None | No changes to the meaning, language, or formatting of the technical content. |
| 10/23/2012 | 0.1 | None | No changes to the meaning, language, or formatting of the technical content. |
| 3/26/2013 | 0.1 | None | No changes to the meaning, language, or formatting of the technical content. |
| 6/11/2013 | 0.1 | None | No changes to the meaning, language, or formatting of the technical content. |
| 8/8/2013 | 0.1 | None | No changes to the meaning, language, or formatting of the technical content. |
| 12/5/2013 | 0.1 | None | No changes to the meaning, language, or formatting of the technical content. |
| 2/11/2014 | 1.0 | Major | Significantly changed the technical content. |
| 5/20/2014 | 2.0 | Major | Significantly changed the technical content. |
| 5/10/2016 | 3.0 | Major | Significantly changed the technical content. |
| 8/16/2017 | 4.0 | Major | Significantly changed the technical content. |
| 5/24/2018 | 5.0 | Major | Significantly changed the technical content. |
| 9/12/2018 | 6.0 | Major | Significantly changed the technical content. |
| 10/16/2019 | 7.0 | Major | Significantly changed the technical content. |
| 6/22/2020 | 8.0 | Major | Significantly changed the technical content. |

| Date | Revision History | Revision Class | Comments |
|---|---|---|---|
| 2/11/2021 | 9.0 | Major | Significantly changed the technical content. |
| 6/10/2021 | 10.0 | Major | Significantly changed the technical content. |
| 11/2/2021 | 11.0 | Major | Significantly changed the technical content. |
| 11/1/2022 | 12.0 | Major | Significantly changed the technical content. |
| 12/20/2022 | 12.1 | Minor | Clarified the meaning of the technical content. |

*[MS-DTSX2-Diff] - v20221220*
*Data Transformation Services Package XML Version 2 File Format*
*Copyright © 2022 Microsoft Corporation*
*Release: December 20, 2022*

# Table of Contents

# 1   Introduction

Version 2 of the Data Transformation Services Package XML (DTSX) file format is an enhanced refactoring of version 1 of DTSX (for more information about version 1, see [MS-DTSX]). DTSX is an XML-based file format that stores the instructions for the processing of a data flow, including transformations and optional processing steps, from its points of origin to its points of destination.

Sections 1.7 and 2 of this specification are normative. All other sections and examples in this specification are informative.

## 1.1   Glossary

This document uses the following terms:

**bcp utility**: A Microsoft SQL Server utility that bulk copies data in a user-specified format between an instance of SQL Server and a data file. The bcp utility can be used to import large numbers of new rows into SQL Server tables or to export data out of tables into data files.

**BSTR**: A binary structure that contains a string. The structure consists of a length prefix, a character string, and a terminator character.

**executable**: A unit of executable code that performs a defined task along the data flow in a package.

**expression**: A combination of symbols (identifiers, literals, functions, and operators) that yields a single data value.

**globally unique identifier (GUID)**: A term used interchangeably with universally unique identifier (UUID) in Microsoft protocol technical documents (TDs). Interchanging the usage of these terms does not imply or require a specific algorithm or mechanism to generate the value. Specifically, the use of this term does not imply or require that the algorithms described in [RFC4122] or [C706] must be used for generating the GUID. See also universally unique identifier (UUID).

**indirect method**: A method by which a configuration file is identified by having a Windows environment variable contain the path to the configuration file, instead of storing the path to the file within the package file itself.

**inferred member**: A member of an Analysis Services dimension whose existence can be inferred from a fact table but which has not yet been loaded from the dimension.

**INI file**: A configuration file that is used by Windows programs to initialize program settings.

**inner variable**: A DTS 2000 variable that is of package scope to a DTS 2000 Package when that package is executed by a parent Microsoft SQL Server Integration Services (SSIS) package. The inner variable is passed in to the DTS 2000 package from a parent Integration Services package.

**language code identifier (LCID)**: A 32-bit number that identifies the user interface human language dialect or variation that is supported by an application or a client computer.

**Microsoft Message Queuing (MSMQ)**: A communications service that provides asynchronous and reliable message passing between distributed applications. In Message Queuing, applications send messages to queues and consume messages from queues. The queues provide persistence of the messages, enabling the sending and receiving applications to operate asynchronously from one another.

**outer variable**: An Integration Services package-scope variable that is passed into a DTS 2000 package. Inside the DTS 2000 package, it becomes an inner variable.

**package file**: A file that contains a collection of tasks that form a workflow that is executed in the order that is defined within the package, with possible conditional branching.

**package parameter**: One of the objects that is in the package. Different values for each parameter can be specified for execution of the package. The value of a parameter can be read by other objects in the package as the package is executing.

**package variable**: A special type of variable within a package file that contains information about the last state of the screen layout of the package. Package variables are used by the user interface when a new session to modify a package is started.

**passive mode**: An FTP mode in which the client initiates a conversation by connection to the FTP server port, and the client also initiates the data communication port connection, to a port specified by the server in its initial response to the client. For more information about FTP active mode and passive mode, see [MSKB-323446].

**precedence constraint**: An object in a Integration Services package that controls the order of execution of tasks or the condition under which a task executes.

**Server Management Objects (SMO) Server**: An instance of SQL Server that is accessed through calls to the Server Management Objects (SMO) object model.

**Simple Mail Transfer Protocol (SMTP)**: A member of the TCP/IP suite of protocols that is used to transport Internet messages, as described in [RFC5321].

**tag table**: A table that is used in the column pattern profile option of the Data Profiling Task. This table contains a list of tags that are used to tag specific terms that are found in the data file.

**task**: The building block of a package. A task consists of code that executes a function, as specified by the options, settings, and parameters of the task that are specified when the task is called.

**TFIDF Score**: One of the options for producing a similarity score used by the Term Extraction Component. The TFIDF score is defined as the TFIDF of a Term T = (frequency of T) * log( (# rows in Input) / (# rows having T) ).

**Transact-SQL**: The Microsoft proprietary version of SQL, the structured query language.

**Unicode**: A character encoding standard developed by the Unicode Consortium that represents almost all of the written languages of the world. The Unicode standard [UNICODE5.0.0/2007] provides three forms (UTF-8, UTF-16, and UTF-32) and seven schemes (UTF-8, UTF-16, UTF-16 BE, UTF-16 LE, UTF-32, UTF-32 LE, and UTF-32 BE).

**universally unique identifier (UUID)**: A 128-bit value. UUIDs can be used for multiple purposes, from tagging objects with an extremely short lifetime, to reliably identifying very persistent objects in cross-process communication such as client and server interfaces, manager entry-point vectors, and RPC objects. UUIDs are highly likely to be unique. UUIDs are also known as globally unique identifiers (GUIDs) and these terms are used interchangeably in the Microsoft protocol technical documents (TDs). Interchanging the usage of these terms does not imply or require a specific algorithm or mechanism to generate the UUID. Specifically, the use of this term does not imply or require that the algorithms described in [RFC4122] or [C706] must be used for generating the UUID.

**variable**: One of the objects that is in a package file. A variable allows a package designer to assign a value to a symbolic name, which can then be accessed by other package objects.

**Windows Management Instrumentation (WMI)**: The Microsoft implementation of Common Information Model (CIM), as specified in [DMTF-DSP0004]. WMI allows an administrator to manage local and remote machines and models computer and network objects using an extension of the CIM standard.

**Windows Presentation Foundation (WPF) unit**: A device-independent unit measure that will render at the same physical size on all devices.

**XPathNavigator**: An object that allows navigation of the set of objects that are returned by an XPATH query.

**MAY, SHOULD, MUST, SHOULD NOT, MUST NOT:** These terms (in all caps) are used as defined in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

## 1.2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the Errata.

References to Microsoft Open Specifications documentation do not include a publishing year because links are to the latest version of the documents, which are updated frequently. References to other documents include a publishing year when one is available.

## 1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information.

[ECMA-327] ECMA, "ECMAScript 3rd Edition Compact Profile", Standard ECMA-327, June 2001, http://www.ecma-international.org/publications/standards/Ecma-327.htm

[ISO/IEC8859-1] ISO/IEC, "Information technology -- 8-bit single-byte coded graphic character sets -- Part 1: Latin alphabet No. 1", ISO/IEC 8859-1:1998, April 1998, http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=28245

**Note** There is a charge to download the specification.

[JavaSpec] Oracle Technology Network, "The Java Language Specification, Third Edition", January 2005, http://docs.oracle.com/javase/specs/jls/se6/html/j3TOC.html

[MS-DTSX] Microsoft Corporation, "Data Transformation Services Package XML File Format".

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, https://www.rfc-editor.org/rfc/rfc2119.html

[RFC2268] Rivest, R., "A Description of the RC2(r) Encryption Algorithm", RFC 2268, March 1998, http://www.rfc-editor.org/rfc/rfc2268.txt

[RFC2459] Housley, R., Ford, W., Polk, W., and Solo, D., "Internet X.509 Public Key Infrastructure Certificate and CRL Profile", RFC 2459, January 1999, http://www.rfc-editor.org/rfc/rfc2459.txt

[UNICODE] The Unicode Consortium, "The Unicode Consortium Home Page", http://www.unicode.org/

[US-ASCII] Injosoft AB, "ASCII Code - The extended ASCII table", https://www.ascii-code.com/

[XML10/5] Bray, T., Paoli, J., Sperberg-McQueen, C.M., et al., Eds., "Extensible Markup Language (XML) 1.0 (Fifth Edition)", W3C Recommendation, November 2008, http://www.w3.org/TR/2008/REC-xml-20081126/

[XMLNS] Bray, T., Hollander, D., Layman, A., et al., Eds., "Namespaces in XML 1.0 (Third Edition)", W3C Recommendation, December 2009, https://www.w3.org/TR/2009/REC-xml-names-20091208/

[XMLSCHEMA1/2] Thompson, H., Beech, D., Maloney, M., and Mendelsohn, N., Eds., "XML Schema Part 1: Structures Second Edition", W3C Recommendation, October 2004, https://www.w3.org/TR/2004/REC-xmlschema-1-20041028/

[XMLSCHEMA2/2] Biron, P., and Malhotra, A., Eds., "XML Schema Part 2: Datatypes Second Edition", W3C Recommendation, October 2004, https://www.w3.org/TR/2004/REC-xmlschema-2-20041028/

[XPATH] Clark, J. and DeRose, S., "XML Path Language (XPath), Version 1.0", W3C Recommendation, November 1999, http://www.w3.org/TR/1999/REC-xpath-19991116/

## 1.2.2 Informative References

[Haselden] Haselden, K., "Microsoft SQL Server 2005 Integration Services", Sams, 2006, ISBN: 0672327813.

[MSDN-ADONETSRC] Microsoft Corporation, "ADO NET Source", https://learn.microsoft.com/en-us/sql/integration-services/data-flow/ado-net-source

[MSDN-BCPU] Microsoft Corporation, "bcp Utility", https://learn.microsoft.com/en-us/previous-versions/sql/sql-server-2008-r2/ms162802(v=sql.105)

[MSDN-C#LS] Microsoft Corporation, "Introduction", C# 7.0 draft specification, https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/language-specification/introduction

[MSDN-CryptProtectData] Microsoft Corporation, "CryptProtectData Function", https://learn.microsoft.com/en-us/windows/desktop/api/dpapi/nf-dpapi-cryptprotectdata

[MSDN-DMXR] Microsoft Corporation, "Data Mining Extensions (DMX) Reference", https://learn.microsoft.com/en-us/sql/dmx/data-mining-extensions-dmx-reference

[MSDN-EPCO] Microsoft Corporation, "Extending Packages with Custom Objects", https://learn.microsoft.com/en-us/sql/integration-services/extending-packages-custom-objects/extending-packages-with-custom-objects

[MSDN-MMASDM] Microsoft Corporation, "Mining Models (Analysis Services - Data Mining)", https://learn.microsoft.com/en-us/sql/analysis-services/data-mining/mining-models-analysis-services-data-mining

[MSDN-MSASDM] Microsoft Corporation, "Mining Structures (Analysis Services - Data Mining)", https://learn.microsoft.com/en-us/sql/analysis-services/data-mining/mining-structures-analysis-services-data-mining

[MSDN-PBCO] Microsoft Corporation, "Performing Bulk Copy Operations", https://learn.microsoft.com/en-us/sql/relational-databases/native-client/features/performing-bulk-copy-operations

[MSDN-POS] Microsoft Corporation, "Processing Options and Settings (Analysis Services)", https://learn.microsoft.com/en-us/sql/analysis-services/multidimensional-models/processing-options-and-settings-analysis-services

[MSDN-TYPCLS] Microsoft Corporation, "Type Class", https://learn.microsoft.com/en-us/dotnet/api/system.type

[MSDN-VBLR] Microsoft Corporation, "Visual Basic Language Reference", https://learn.microsoft.com/en-us/dotnet/visual-basic/language-reference/index

[MSDN-VBSLR] Microsoft Corporation, "VBScript Language Reference", https://learn.microsoft.com/en-us/previous-versions//d1wf56tt(v=vs.85)

[MSFT-CDC] Microsoft Corporation, "About Change Data Capture (SQL Server)", https://learn.microsoft.com/en-us/sql/relational-databases/track-changes/about-change-data-capture-sql-server

[SCHNEIER] Schneier, B., "Applied Cryptography, Second Edition", John Wiley and Sons, 1996, ISBN: 0471117099, http://www.wiley.com/WileyCDA/WileyTitle/productCd-0471117099.html

[XMLEncSynProc] Imamura, T., Dillaway, B., Simon, E., et al., "XML Encryption Syntax and Processing Version 1.1", W3C Recommendation, April 2013, http://www.w3.org/TR/xmlenc-core/

## 1.3 Overview

This document describes the persistence file structure for Integration Services. Integration Services is used to orchestrate activities to move and conform data from disparate sources in an enterprise into a convenient location and shape, often for analysis and reporting. The DTSX structure describes these orchestration activities as a pathway that is comprised of a set of steps, each of which can involve processing of the data as it flows through the step. Steps can be conditional or iterative, and not all of the steps that are described in this document are necessarily executed on any one passing of data through the defined steps.

The set of steps starts with a reference to a source from which to import data to begin the set of operations that the data will be passed through. The DTSX structure then allows the user to specify that the data is to pass through an executable that performs a processing task upon the data, or that the data is to pass through a pipeline component that modifies and/or transforms the data. The structure also allows the ability to define a pathway for the data with conditional elements that can change the data flow pathways or operate iteratively in a loop.

Finally, the structure allows the ability to define a final destination for the post-processed data flow. A file that conforms to the DTSX structure as defined in this document can then be processed by the data transformation processing engine, or it can be opened in a UI to make modifications to the definition.

The following diagram shows one typical use of Integration Services: to consolidate a data flow from multiple sources, and while consolidating, also transform and assure consistency of all the data.



**Figure 1: Integration Services package data flow**

The intermediate step in the data flow can be one of the following:

- **Executable Task**: Performs a defined type of work along the data flow. Custom tasks can be written by end users to perform any task that is not included with the Integration Services product. Examples of standard tasks that ship with Integration Services are tasks that manipulate

an external file, process Microsoft SQL Server Analysis Services objects, run a data mining query, perform file transfers with FTP, perform a database backup, and execute an external executable.

- **Component**: Components perform data transformations along the data flow. Examples of components that ship with Integration Services are components that perform aggregation; create columns derived by a specified formula; and perform row counts, sampling, lookups, and many other tasks. End users can also create custom components to perform additional transformations that are not included with the standard product.

The following diagram is a detailed example of a data flow described by the DTSX structure format. The diagram shows the presence of two data sources (an OLE DB Source and a Flat File Source) and three final data destinations (a Raw File Destination, an OLE DB destination, and a SQL Server Destination). The steps in the data flow, based on conditional processing, determine which destination the data ultimately flows to and how it is transformed and processed on the route to its final destination.

Some of the intermediate steps are conditional. For example, in the diagram there are two arrows between Lookup and Fuzzy Lookup. One arrow represents the data flow for erroneous data, and the other represents the flow for standard output.
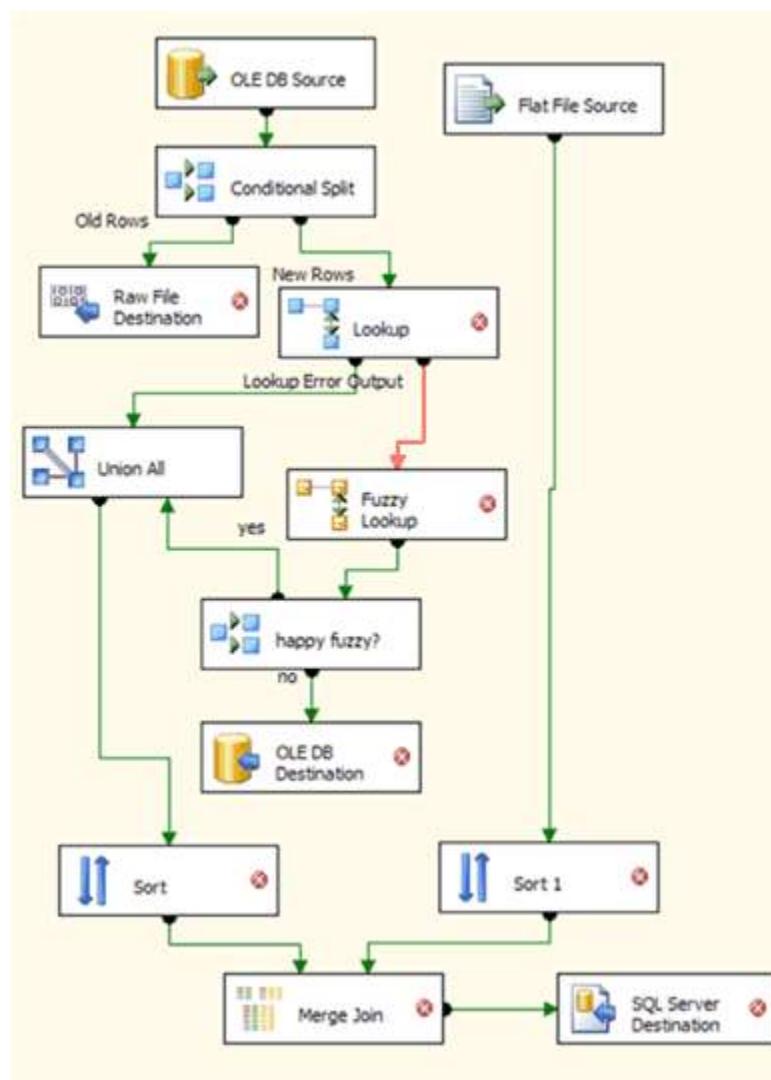


**Figure 2: Data flow described by DTSX structure**

**Differences between DTSX and DTSX 2.0**

DTSX version 2.0 (DTSX 2.0) is an enhanced refactoring of the XML that is documented in [MS-DTSX]. Both [MS-DTSX] and [MS-DTSX2] describe the package files that are used by Integration Services. DTSX 2.0 is used for package files starting with Microsoft SQL Server 2012.

DTSX 2.0 has the following XML refactoring changes from DTSX:

▪ DTSX has **Property** elements with a **Name** attribute to specify the property name. In DTSX 2.0, named properties are XML attributes of the parent element. However, a small number of named **Property** elements remain in DTSX 2.0.

▪ Some new attributes have been added to DTSX 2.0.

▪ To organize the larger number of attributes in DTSX 2.0, **AttributeGroup** groups of attributes have been defined, and are referenced by the complex type definitions in the XSD of DTSX 2.0.

▪ DTSX 2.0 specifies defaults for most attribute values, and sets the XSD **usage** attribute to "optional" for attributes with defaults. In DTSX, all property elements are always present, even if the element value is equal to the default value.

▪ In places where repeated element instances were allowed in DTSX, these repeated elements have been placed inside a parent element to hold the collection in DTSX 2.0, in most cases. For example, if multiple Executable elements appear within a package file, the **Executable** elements are contained in a parent **Executables** element in DTSX 2.0.

## 1.4   Relationship to Protocols and Other Structures

The usual structure format is clear-text XML [XML10/5].

## 1.5   Applicability Statement

This document describes an XML-based file persistence format that is used to describe a data flow that can be processed by a processing engine.

## 1.6   Versioning and Localization

**Structure Versions:**  This document specifies the structures for the following schema versions of version 2 of the Data Transformation Services Package XML (DTSX2) file format:<1>

▪ DTSX2 schema for version 2012/01 (DTSX2 2012/01)

▪ DTSX2 schema for version 2014/01 (DTSX2 2014/01)

The structures and properties in this document apply to all schema versions, unless otherwise indicated. That is, if a particular structure or property applies only to a specific schema version, that schema version is identified in the relevant section.

The XML Schema definitions (XSDs) for the two DTSX2 schema versions are provided in section 5.1.

**Localization:**  There are no localization-dependent structures in the DTSX2 file format.

## 1.7   Vendor-Extensible Fields

The DTSX structure allows for vendor extensibility. Vendors can add custom-programmed objects of the following types to the system and, therefore, to the DTSX structure:

- Executable

- ConnectionManager

- Log provider

- ForEachEnumerator

- Component

The following table lists the affected elements, attributes, properties, or types for each of the above cases.

| Custom item | Reference | Elements, attributes, properties, or types |
|---|---|---|
| Executable | Custom Executable | **ObjectData** element |
| ConnectionManager | Custom ConnectionManager | **ObjectData** element |
| LogProvider | Custom Log Provider | None |
| ForEachEnumerator | Custom ForEachEnumerator | **ObjectData** element |
| Component | Custom Component | <ul><li>PipelineComponentComponentClassIDEnum type</li><li>PipelineComponentPropertyNameEnum type</li><li>PipelineComponentInputPropertyNameEnum type</li><li>PipelineComponentOutputPropertyNameEnum type</li><li>PipelineComponentInputColumnPropertyNameEnum type</li><li>PipelineComponentOutputColumnPropertyNameEnum type</li></ul> |

# 2    Structures

This section contains the definition of the Data Transformation Services Package XML (DTSX) structure. DTSX structure instances are contained in a single file called a package file.

The usual structure format is clear-text XML [XML10/5]. The XML MAY be encrypted.<2> XML Schema definition language (XSD) is used to describe the DTSX structure [XMLSCHEMA1/2] [XMLSCHEMA2/2].

## 2.1    Additional Normative Requirements Concerning the XSD Fragments Provided

### 2.1.1    Constraints beyond the XSD Definitions for Complex Types

Each section of the structure definition has an accompanying XSD to define the type for a particular element of the structure. However, in some cases, there are additional constraints on what can be contained in a defined element that go beyond what is expressed in the XSD. These constraints MUST be followed. Any such constraints for each complex type are noted in the tables and information that accompany the XSD within each section.

### 2.1.2    <xs:sequence> Usage Expressed in the XSD is Optional

Throughout this structure definition, the <xs:sequence> XSD construct is used.<3>

## 2.2    Namespaces

This specification defines and references various XML namespaces using the mechanisms that are specified in [XMLNS]. Although this specification associates a specific XML namespace prefix for each XML namespace that is used, the choice of any particular XML namespace prefix is implementation-specific and is not significant for interoperability.

| Prefix | Namespace URI | Reference |
|---|---|---|
| xs | http://www.w3.org/2001/XMLSchema | [XMLSCHEMA1/2] [XMLSCHEMA2/2] |
| xsd | http://www.w3.org/2001/XMLSchema | [XMLSCHEMA1/2] [XMLSCHEMA2/2] |
| DTS | www.microsoft.com/SqlServer/Dts | Section 5.1 |
| WSTask | www.microsoft.com/sqlserver/dts/tasks/webservicetask | Section 5.2 |
| MessageQueueTask | www.microsoft.com/sqlserver/dts/tasks/messagequeuetask | Section 5.3 |
| SendMailTask | www.microsoft.com/sqlserver/dts/tasks/sendmailtask | Section 5.4 |
| ActiveXScriptTask (applies to DTSX2 2012/01) | www.microsoft.com/sqlserver/dts/tasks/activexscripttask | Section 5.5 |
| Exec80PackageTask (applies to DTSX2 2012/01) | www.microsoft.com/sqlserver/dts/tasks/exec80packagetask | Section 5.6 |
| BulkInsertTask | www.microsoft.com/sqlserver/dts/tasks/bulkinserttask | Section 5.7 |
| SQLTask | www.microsoft.com/sqlserver/dts/tasks/sqltask | Section 5.8 |

## 2.3    Executable Element

All Microsoft SQL Server Integration Services (SSIS) packages MUST contain a single root element. This element is the **Executable** element. The root **Executable** element MUST be of type **DTS:ExecutableTypePackage**.

The following is the XSD for the **Executable** root element.

```xml
<xs:element name="Executable" type="DTS:ExecutableTypePackage"/>
```

## 2.4    ExecutableTypePackage

The **ExecutableTypePackage** complex type defines the content of the **Executable** root element. For more details, see section 2.3.

The following is the XSD definition for the **ExecutableTypePackage** complex type.

```xml
<xs:complexType name="ExecutableTypePackage">
  <xs:sequence>
    <xs:element name="Property" maxOccurs="unbounded">
      <xs:complexType>
        <xs:complexContent>
          <xs:extension base="DTS:PropertyElementBaseType">
            <xs:attribute name="Name"
                type="DTS:ExecutableTypePackagePropertyNameEnum"
                use="required"/>
          </xs:extension>
        </xs:complexContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="ConnectionManagers" type="DTS:ConnectionManagersType"
                minOccurs="0"/>
    <xs:element name="Configurations" type="DTS:ConfigurationsType"
                minOccurs="0"/>
    <xs:element name="LogProviders" type="DTS:LogProvidersType"
                minOccurs="0"/>
    <xs:element name="Variables" type="DTS:VariablesType"
                minOccurs="0"/>
    <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType" />
    <xs:element name="PropertyExpression"
                type="DTS:PropertyExpressionElementType"
                minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Executables" type="DTS:ExecutablesType"
                minOccurs="0"/>
    <xs:element name="PrecedenceConstraints" minOccurs="0"
                type="DTS:PrecedenceConstraintsType"/>
    <xs:element name="EventHandlers" type="DTS:EventHandlersType"
                minOccurs="0"/>
    <xs:element name="PackageVariables" type="DTS:PackageVariablesType"
                minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="PackageParameters" type="DTS:PackageParametersType" minOccurs="0"
                maxOccurs="1" />
  </xs:sequence>
  <xs:attribute name="ExecutableType" use="required"
            fixed="SSIS.Package.2" />
  <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
  <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
  <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
  <xs:attributeGroup ref="DTS:ExecutableTypePackageAttributeGroup"/>
  <xs:attribute name="refId" type="xs:string" use="required"
                form="qualified"/>
</xs:complexType>
```

The following table provides additional information about the elements and types in the **ExecutableTypePackage** complex type.

| Element | Constraints | Type definition | Description |
|---------|-------------|-----------------|-------------|
| Property | In some cases, constraints are placed on the **Property** element value for specific values of the **Name** attribute of the **Property** element. If such constraints exist, they are specified where the meaning of the particular property name is specified. These constraints MUST be followed. If the optional **DataType** attribute is present, the value of the **Property** element MUST be of the type indicated by the **DataType** attribute. | Inline in the XSD in this section. | Specifies values for many different properties that are associated with an instance of the containing **Executable** element, according to the **Name** attribute of each **Property** element. |
| ConnectionManagers | None | ConnectionManagersType | Specifies a collection of elements of type **ConnectionManagerType**. Each such element defines a connection that is available to the package. |
| Configurations | None | ConfigurationsType | Specifies a collection of elements of type **ConfigurationType**. Each such element defines a connection that is available to the package. |
| LogProviders | None | LogProvidersType | Specifies a collection of elements of type **LogProviderType**. Each such element defines a log provider that is available to the package. |
| Variables | None | VariablesType | Specifies a collection of elements of type **VariableType**. Each such element defines a variable that is available to the package. |
| LoggingOptions | None | LoggingOptionsType | Specifies information about what kind of information is logged for this package and where it is recorded. |
| PropertyExpression | Additional constraints exist for the **PropertyExpression** element. For a description of these constraints, see PropertyExpressionElement Type. | PropertyExpressionElement Type | Serves the same purpose as specified for the **Property** element except that an expression that is evaluated at run time is contained in the element's value. |

| Element | Constraints | Type definition | Description |
|---|---|---|---|
| Executables | None | ExecutablesType | Specifies a collection of elements of type **AnyNonPackageExecutable Type**. Each such element defines an executable that is available to the package. |
| PrecedenceConstra ints | None | PrecedenceConstraintsType | Specifies a collection of elements of type **PrecedenceConstraintType**. Each such element defines a precedence constraint that is available to the package. |
| EventHandlers | None | EventHandlersType | Specifies a collection of elements of type **EventHandlerType**. Each such element defines an event handler that is available to the package. |
| PackageVariables | None | PackageVariablesType | Specifies a collection of elements of type **PackageVariableType**. Each such element defines a package variable that is available to the package. |
| PackageParameter s | None | PackageParametersType | Specifies a collection of elements of type **PackageParameterType**. Each such element defines a package parameter that is available to the package. |

The following table provides additional information about the attributes, attribute groups, and types for the **ExecutableTypePackage** type.

| Attribute | Description |
|---|---|
| ExecutableType | Specifies information about the contents of the **Executable** element on which the attribute value appears. |
| AllExecutableAttributeGroup | An attribute group that specifies attributes that are common to many different types throughout the schema. |
| BaseExecutablePropertyAttributeGroup | An attribute group that specifies attributes that are common to many different types throughout the schema. |
| BasePropertyAttributeGroup | An attribute group that specifies attributes that are common to many different types throughout the schema. |
| ExecutableTypePackageAttributeGroup | An attribute group that specifies attributes for the **ExecutableTypePackage** type. |
| refId | A string value that specifies a unique **refId** for this object. Elsewhere within the containing executable or within the package, the value of **refId** can be used to refer to this object.<4> |

### 2.4.1 PackageVariablesType

The **PackageVariablesType** complex type is the container type for a collection of elements of type PackageVariableType.

The following is the XSD of the **PackageVariablesType** type.

```
<xs:complexType name="PackageVariablesType">
  <xs:sequence>
    <xs:element name="PackageVariable" type="DTS:PackageVariableType"
                minOccurs="0" maxOccurs="unbounded" form="qualified"/>
  </xs:sequence>
</xs:complexType>
```

The following table provides additional information about the elements, types, and constraints for the **PackageVariablesType** complex type.

| Element | Type definition | Description |
|---|---|---|
| PackageVariable | PackageVariableType | Specifies a **PackageVariable** element that contains the values and content for a package variable. |

### 2.4.1.1 PackageVariableType

The **PackageVariableType** complex type is used to specify package variables. Package variables are used to save screen information so that the Integration Services UI editor can present the screen for editing in the same state as it was left in.

The following is the XSD for the **PackageVariableType** complex type.

```
<xs:complexType name="PackageVariableType">
  <xs:sequence>
    <xs:element name="Property"
                maxOccurs="unbounded">
      <xs:complexType>
        <xs:complexContent>
          <xs:extension base="DTS:PropertyElementBaseType">
            <xs:attribute name="Name"
                type="DTS:PackageVariablePropertyNameEnum"
                use="required"/>
          </xs:extension>
        </xs:complexContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="PropertyExpression"
                type="DTS:PropertyExpressionElementType"
                minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

The following table provides additional information about the elements, types, and constraints in the **PackageVariableType** complex type.

| Element | Additional constraints | Type definition | Description |
|---|---|---|---|
| Property | In some cases, constraints are placed on the **Property** element value for specific | Inline in the XSD in this section. | **Property** elements contain values for many different properties that |

| Element | Additional constraints | Type definition | Description |
|---|---|---|---|
| | values of the **Name** attribute of the **Property** element. If such constraints exist, they are specified where the meaning of the particular property name is specified. These constraints MUST be followed. If the optional **DataType** attribute is present, the value of the **Property** element MUST be of the type that is specified by the **DataType** attribute. | | are associated with an instance of the containing **Executable** element, according to the **Name** attribute of each **Property** element. |
| PropertyExpression | Additional constraints exist for the **PropertyExpression** element. | PropertyExpressionElementType | The **PropertyExpression** element serves the same purpose as specified for the **Property** element, except that an expression that is evaluated at run time is contained in the element's value that is stored in the Integration Services file format. |

The following table provides additional information about the attributes and types for the **Property** element that is declared inline in the complex type declaration. The **Property** element is an extension of the **PropertyElementBaseType** type with the addition of the **Name** attribute, which is specified in the following table.

| Attribute | Constraints | Description |
|---|---|---|
| Name | None | Specifies the name of the property. Each named property has its own use and interpretation. The meaning of each named property is given with the definition of the type for the **Name** attribute. |

### 2.4.1.1.1 PackageVariablePropertyNameEnum

The **PackageVariablePropertyNameEnum** type contains the values for the **Name** attribute of the property elements that are allowed for the PackageVariableType complex type.

The following is the XSD for the **PackageVariablePropertyNameEnum** type.

```
<xs:simpleType name="PackageVariablePropertyNameEnum">
  <xs:union memberTypes="DTS:BasePropertyNameEnum">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="PackageVariableValue"/>
        <xs:enumeration value="PackageVariable"/>
        <xs:enumeration value="Namespace"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:union>
</xs:simpleType>
```

The following table specifies the enumeration values for the **PackageVariablePropertyNameEnum** type. The **PackageVariablePropertyNameEnum** type is a union of the **BasePropertyNameEnum** type with the values that are specified in the following table. The values described in this table do not include the values already specified for the **BasePropertyNameEnum** type.

| Enumeration value | Description |
|---|---|
| PackageVariableValue | The **PackageVariableValue** enumeration value specifies that the element contains text that specifies screen locations for visual design of packages. It is not in the scope of this document to describe the interpretation of this text. |
| PackageVariable | The **PackageVariable** enumeration value is a synonym for **PackageVariableValue**. |
| Namespace | The **Namespace** enumeration value specifies that the element value contains an additional qualifier that enables unique identification of a package variable. Two package variables that have the same name can be distinguished by having a different **Namespace** value. |

## 2.4.2 ConfigurationsType

The **ConfigurationsType** complex type is the container type for a collection of elements of type ConfigurationType.

The following is the XSD of the **ConfigurationsType** type.

```
<xs:complexType name="ConfigurationsType">
  <xs:sequence>
    <xs:element name="Configuration" type="DTS:ConfigurationType"
              minOccurs="0" maxOccurs="unbounded" form="qualified"/>
  </xs:sequence>
</xs:complexType>
```

The following table provides additional information about the elements, types, and constraints for the **ConfigurationsType** complex type.

| Element | Type definition | Description |
|---|---|---|
| Configuration | ConfigurationType | Specifies a **Configuration** element, which contains the options, parameters, and settings for a configuration. |

## 2.4.2.1 ConfigurationType

The **ConfigurationType** complex type contains configuration information that is accessed by other elements within the package.

The following is the XSD for the **ConfigurationType** complex type.

```
<xs:complexType name="ConfigurationType">
  <xs:sequence>
    <xs:element name="PropertyExpression"
              type="DTS:PropertyExpressionElementType"
              minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
  <xs:attributeGroup ref="DTS:ConfigurationAttributeGroup"/>
```

```
     </xs:complexType>
```

The following table provides additional information about the elements, types, and constraints in the **ConfigurationType** complex type.

| Element | Additional constraints | Type definition | Description |
|---|---|---|---|
| PropertyExpression | Additional constraints exist for the **PropertyExpression** element. | PropertyExpressionElementType | The **PropertyExpression** element serves the same purpose as described for the **Property** element, except that an expression that is evaluated at run time is contained in the element's value that is stored in the Integration Services file format. |

The following table provides additional information about the attributes, attribute groups, and types for the **ConfigurationType** type.

| Attribute | Description |
|---|---|
| BasePropertyAttributeGroup | An attribute group that specifies attributes that are common to many different types throughout the schema. |
| ConfigurationAttributeGroup | An attribute group that specifies attributes for the **ConfigurationType** type. |

## 2.4.2.1.1 ConfigurationAttributeGroup

The **ConfigurationAttributeGroup** attribute group contains the attributes for the ConfigurationType type.

The following is the XSD for the **ConfigurationAttributeGroup** attribute group.

```
    <xs:attributeGroup name="ConfigurationAttributeGroup">
      <xs:attribute name="ConfigurationType" type="DTS:ConfigurationTypeEnum"
                    default="0" use="optional" form="qualified"/>
      <xs:attribute name="ConfigurationString" type="xs:string" default=""
                    use="optional" form="qualified"/>
      <xs:attribute name="ConfigurationVariable" type="xs:string" default=""
                    use="optional" form="qualified"/>
    </xs:attributeGroup>
```

The following table specifies the attributes for the **ConfigurationAttributeGroup** attribute group.

| Attribute | Description |
|---|---|
| ConfigurationType | An enumeration value that specifies the source from which the configuration is read. |
| ConfigurationString | A string value that is used to institute the configuration. The use of the string varies with the setting for **ConfigurationType**. |
| ConfigurationVariable | A string value that specifies the full path, within the package, to the property that is being configured. |

#### 2.4.2.1.1.1 ConfigurationTypeEnum

The **ConfigurationTypeEnum** type contains the enumeration values that specify the sources from which a configuration can be read.

The following is the XSD of the **ConfigurationTypeEnum** type.

```
<xs:simpleType name="ConfigurationTypeEnum">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="10"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the descriptions for the enumeration values of the **ConfigurationTypeEnum** type.

| Enumeration value | Description |
|---|---|
| 0 | A parent variable. |
| 1 | A configuration file. |
| 2 | An environment variable. |
| 3 | A registry entry. |
| 4 | A parent variable, indirect method. |
| 5 | A configuration file, indirect method. |
| 6 | A registry entry, indirect method. |
| 7 | A database table. |
| 8 | An INI file. |
| 9 | A database server, indirect method. |
| 10 | An INI file, indirect method. |

### 2.4.3 LogProvidersType

The **LogProvidersType** complex type is the container type for a collection of elements of type LogProviderType.

```
<xs:complexType name="LogProvidersType">
  <xs:sequence>
    <xs:element name="LogProvider" type="DTS:LogProviderType"
                minOccurs="0" maxOccurs="unbounded"  form="qualified"/>
  </xs:sequence>
</xs:complexType>
```

The following table provides additional information about the elements, types, and constraints for the **LogProvidersType** complex type.

| Element | Type definition | Description |
|---|---|---|
| LogProvider | LogProviderType | Specifies a **LogProvider** element, which contains the options, parameters, and settings for a log provider. |

### 2.4.3.1 LogProviderType

The **LogProviderType** complex type is used to define a particular log provider that is used by the package.

The following is the XSD for the **LogProviderType** complex type.

```
<xs:complexType name="LogProviderType" >
  <xs:sequence>
    <xs:element name="PropertyExpression"
                type="DTS:PropertyExpressionElementType"
                minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="ObjectData" type="DTS:LogProviderObjectDataType"/>
  </xs:sequence>
  <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
  <xs:attributeGroup ref="DTS:LogProviderAttributeGroup"/>
</xs:complexType>
```

The following table provides additional information about the elements, types, and constraints in the **LogProviderType** complex type.

| Element | Additional constraints | Type definition | Description |
|---|---|---|---|
| PropertyExpression | Additional constraints exist for the **PropertyExpression** element. | PropertyExpressionElementType | The **PropertyExpression** element serves the same purpose as specified for the **Property** element, except that an expression that is evaluated at run time is contained in the element's value that is stored in the Integration Services file format. |
| ObjectData | None | LogProviderObjectDataType | Contains information that is specific to the **LogProvider**. For more information, see Executable ObjectData Types. |

The following table provides additional information about the attributes, attribute groups, and types for the **LogProviderType** type.

| Attribute | Description |
|---|---|
| BasePropertyAttributeGroup | An attribute group that specifies attributes that are common to many different types throughout the schema. |
| LogProviderAttributeGroup | An attribute group that specifies attributes for the **LogProviderType** type. |

### 2.4.3.1.1 LogProviderAttributeGroup

The **LogProviderAttributeGroup** attribute group contains the attributes for the **LogProviderAttributeGroup** type.

The following is the XSD for the **LogProviderAttributeGroup** attribute group.

```
<xs:attributeGroup name="LogProviderAttributeGroup">
  <xs:attribute name="DelayValidation" type="DTS:BooleanStringCap"
                default="False" use="optional" form="qualified"/>
  <xs:attribute name="ConfigString" type="xs:string" default=""
                use="optional" form="qualified"/>
</xs:attributeGroup>
```

The following table specifies the attributes for the **LogProviderAttributeGroup** attribute group.

| Attribute | Description |
|---|---|
| DelayValidation | A value of type **DTS:BooleanStringCap** that specifies whether validation is delayed until the Executable element that contains this LogProvider executes. <br> "True": The validation of the **LogProvider** is delayed until the container is executed. <br> "False": The validation of the **LogProvider** is not delayed until run time. |
| ConfigString | A string value. This value is used by each specific log provider in a different way. |

### 2.4.3.1.2 LogProviderObjectDataType

The **LogProviderObjectDataType** complex type provides information that is specific to a particular instance of a **LogProvider** element.

The following is the XSD for the **LogProviderObjectDataType** complex type.

```
<xs:complexType name="LogProviderObjectDataType">
  <xs:sequence>
    <xs:element name="InnerObject" form="unqualified"
                type="DTS:InnerObjectObjectDataType"/>
  </xs:sequence>
</xs:complexType>
```

| Element | Additional constraints | Type definition | Description |
|---|---|---|---|
| InnerObject | None | Inline in the XSD in this section. | Can contain additional information specific to an instance of a **LogProvider** element.<5> |

## 2.4.4 ConnectionManagersType

The **ConnectionManagersType** complex type is the container type for a collection of elements of type ConnectionManagerType.

The following is the XSD of the **ConnectionManagersType** complex type.

```
<xs:complexType name="ConnectionManagersType">
  <xs:sequence>
    <xs:element name="ConnectionManager" type="DTS:ConnectionManagerType"
                minOccurs="0" maxOccurs="unbounded" form="qualified"/>
  </xs:sequence>
</xs:complexType>
```

The following table provides additional information about the elements, types, and constraints for the **ConnectionManagersType** complex type.

| Element | Type definition | Description |
|---|---|---|
| ConnectionManager | ConnectionManagerType | Specifies a **ConnectionManager** element, which contains the options, parameters, and settings for a connection manager. |

## 2.4.4.1 ConnectionManagerType

The **ConnectionManagerType** complex type is used to specify a connection to a data source.

The following is the XSD of the **ConnectionManagerType** complex type.

```
<xs:complexType name="ConnectionManagerType">
  <xs:sequence>
    <xs:element name="PropertyExpression"
                type="DTS:PropertyExpressionElementType"
                minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="ObjectData"
                type="DTS:ConnectionManagerObjectDataType"/>
  </xs:sequence>
  <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
  <xs:attributeGroup ref="DTS:ConnectionManagerAttributeGroup"/>
  <xs:attribute name="refId" type="xs:string" use="required"
                form="qualified"/>
</xs:complexType>
```

The following table provides additional information about the elements, types, and constraints in the **ConnectionManagerType** complex type.

| Element | Additional constraints | Type definition | Description |
|---|---|---|---|
| PropertyExpression | Additional constraints exist for the **PropertyExpression** element. | PropertyExpressionElementType | The **PropertyExpression** element serves the same purpose as specified for the **Property** element, except that an expression that is evaluated at run time is contained in the element's value that is stored in the Integration |

| Element | Additional constraints | Type definition | Description |
|---|---|---|---|
| | | | Services file format. Properties contained in the **ObjectData** element can also become **PropertyExpressions** in this location. |
| ObjectData | The contents of the **ObjectData** element vary with the data source that the **ConnectionManager** element is specifying. | ConnectionManagerObjectDataType | Contains information that is specific to the containing **ConnectionManager** element. The nature of the information varies with the type of data that the connection manager is connecting to, which is specified by the **CreationName** property. |

The following table provides additional information about the attributes, attribute groups, and types for the **ConnectionManagerType** complex type.

| Attribute | Description |
|---|---|
| BasePropertyAttributeGroup | An attribute group that specifies attributes that are common to many different types throughout the schema. |
| ConnectionManagerAttributeGroup | An attribute group that specifies attributes for the **ConnectionManagerType** type. |
| refId | A string value that specifies a unique **refId** for this object. Elsewhere within the containing executable or within the package, the value of **refId** can be used to refer to this object.<6> |

### 2.4.4.1.1 ConnectionManagerAttributeGroup

The **ConnectionManagerAttributeGroup** attribute group contains the attributes for the ConnectionManagerType type.

The following is the XSD for the **ConnectionManagerAttributeGroup** attribute group.

```
<xs:attributeGroup name="ConnectionManagerAttributeGroup">
  <xs:attribute name="DelayValidation" type="DTS:BooleanStringCap"
                default="False" use="optional" form="qualified"/>
</xs:attributeGroup>
```

The following table specifies the attributes for the **ConnectionManagerAttributeGroup** attribute group.

| Attribute | Description |
|---|---|
| DelayValidation | A value of type **DTS:BooleanStringCap** that specifies whether or not validation is delayed until the container of this **ConnectionManager** executes. "False" specifies that the validation of the ConnectionManager is not delayed until run time. "True" specifies that the validation of the ConnectionManager is delayed until run time. |

### 2.4.4.1.2 ConnectionManagerObjectDataType

The **ConnectionManagerObjectDataType** complex type is used to contain additional information that is specific to each type of data connection that is not expressed in the **Property** elements of a **ConnectionManager** instance.

The following is the XSD of the **ConnectionManagerObjectDataType** complex type.

```
<xs:complexType name="ConnectionManagerObjectDataType">
    <xs:choice>
      <xs:element name="ConnectionManager" type=
       "DTS:ConnectionManagerObjectDataConnectionManagerType"/>
      <xs:element name="MsmqConnectionManager" type=
       "DTS:ConnectionManagerObjectDataMsmqConnectionManagerType"
                   form="unqualified"/>
      <xs:element name="SMOServerConnectionManager" type=
       "DTS:ConnectionManagerObjectDataSMOServerConnectionManagerType"
                   form="unqualified"/>
      <xs:element name="SmtpConnectionManager" type=
       "DTS:ConnectionManagerObjectDataSmtpConnectionManagerType"
                   form="unqualified"/>
      <xs:element name="WmiConnectionManager" type=
       "DTS:ConnectionManagerObjectDataWmiConnectionManagerType"
                   form="unqualified"/>
    </xs:choice>
</xs:complexType>
```

The following table provides additional information about the elements, types, and constraints in the **ConnectionManagerType** complex type.

| Element | Additional constraints | Type definition | Description |
|---|---|---|---|
| ConnectionManager | Depending on the type of the connection, there are limits on what can be placed in the **ConnectionManager** element. These constraints MUST be followed. Depending on the type of the connection, there are constraints on the allowed values for the attributes that are contained in the ConnectionManagerConnectionManagerAttributeGroup attribute group. These constraints MUST be followed. | ConnectionManagerObjectDataConnectionManagerType | Contains information for the following connection types: OLE DB Flat file ADO.NET Analysis Services File Cache Excel FTP HTTP Multi-file |

| Element | Additional constraints | Type definition | Description |
|---|---|---|---|
| | | | Multi-flat file<br><br>Open Database Connectivity (ODBC)<br><br>Microsoft SQL Server Mobile |
| MsmqConnectionManager | None | ConnectionManagerObjectDataMsmqConnectionManagerType | Contains information that is specific to the MSMQ **Connection Manager** type. |
| SMOServerConnectionManager | None | ConnectionManagerObjectDataSMOServerConnectionManagerType | Contains information that is specific to the Server Management Objects (SMO) Server **Connection Manager** type. |
| SmtpConnectionManager | None | ConnectionManagerObjectDataSmtpConnectionManagerType | Contains information that is specific to the Simple Mail Transfer Protocol (SMTP) **Connection Manager** type. |
| WmiConnectionManager | None | ConnectionManagerObjectDataWmiConnectionManagerType | Contains information that is specific to the Windows Management Instrumentation (WMI) **Connection Manager** type. |

#### 2.4.4.1.2.1   ConnectionManagerObjectDataConnectionManagerType

The **ObjectData** element in the **ConnectionManager** contains a **ConnectionManager** element of type **ConnectionManagerObjectDataConnectionManagerType** when the data source is one of the following:

- OLE DB

- Flat file

- ADO.NET

- Analysis Services

- File

- Cache

- Excel

- FTP

- HTTP

- Multi-file

- Multi-flat file

- ODBC

- SQL Server Mobile

The following is the XSD of the **ConnectionManagerObjectDataConnectionManagerType** type.

```
<xs:complexType name="ConnectionManagerObjectDataConnectionManagerType">
  <xs:sequence>
      <xs:choice minOccurs="0">
        <xs:element name="FlatFileColumns" type="DTS:FlatFileColumnsType"/>
        <xs:element name="CacheColumns" type="DTS:CacheColumnsType"
                maxOccurs="unbounded"/>
        <xs:element name="FtpConnection" type="DTS:FtpConnectionType"/>
        <xs:element name="HttpConnection" type="DTS:HttpConnectionType"/>
      </xs:choice>
  </xs:sequence>
  <xs:attributeGroup
   ref="DTS:ConnectionManagerConnectionManagerAttributeGroup"/>
</xs:complexType>
```

The following table provides additional information about the elements, types, and constraints in the **ConnectionManagerType** complex type.

| Element | Additional constraints | Type definition | Description |
|---------|------------------------|-----------------|-------------|
| FlatFileColumns | The **FlatFileColumns** element MUST NOT appear unless the data source for the **ConnectionManager** is a flat file or is multiple flat files. The **FlatFileColumns** element MUST be included if the data source for the **ConnectionManager** is a flat file or is multiple flat files. | FlatFileColumnsType | Contains the definition of a flat file column that is present in the data source. |
| CacheColumns | The **CacheColumns** element MUST NOT appear unless the data source for the **ConnectionManager** is a Cache data source. | CacheColumnsType | Contains the definition of a cache column that |

| Element | Additional constraints | Type definition | Description |
|---|---|---|---|
| | The **CacheColumns** element MUST be included if the data source is Cache data. | | is present in the data source. |
| FtpConnection | The **FtpConnection** element MUST NOT appear unless the data source for the **ConnectionManager** is an FTP data source. The **FtpConnection** element MUST be included if the data source is an FTP data source. | FtpConnectionType | Contains the properties of an FTP connection. |
| HttpConnection | The **HttpConnection** element MUST NOT appear unless the data source for the **ConnectionManager** is an HTTP data source. The **HttpConnection** element MUST be included if the data source is an HTTP data source. | HttpConnectionType | Contains the properties of an HTTP connection. |

The following table provides additional information about the attributes, attribute groups, and types for the **ConfigurationType** type.

| Attribute | Description |
|---|---|
| ConnectionManagerConnectionManagerAttributeGroup | An attribute group that specifies attributes for the **ConnectionManagerObjectDataConnectionManagerType** type. |

### 2.4.4.1.2.1.1 ConnectionManagerConnectionManagerAttributeGroup

The **ConnectionManagerConnectionManagerAttributeGroup** attribute group contains the attributes for the ConnectionManagerObjectDataConnectionManagerType type.

The following is the XSD for the **ConnectionManagerConnectionManagerAttributeGroup** attribute group.

```
<xs:attributeGroup name="ConnectionManagerConnectionManagerAttributeGroup">
  <xs:attribute name="Retain" type="DTS:BooleanStringCap" default="False"
                use="optional" form="qualified"/>
  <xs:attribute name="ConnectionString" type="xs:string" use="required"
                form="qualified"/>
  <xs:attribute name="FileUsageType" type="DTS:FileUsageTypeEnum"
                default="0" use="optional" form="qualified"/>
  <xs:attribute name="Format" type="DTS:FormatEnum" use="optional"
                form="qualified"/>
  <xs:attribute name="LocaleID" type="xs:int" use="optional"
                form="qualified"/>
  <xs:attribute name="Unicode" type="DTS:BooleanStringCap" default="False"
                use="optional" form="qualified"/>
  <xs:attribute name="HeaderRowsToSkip" type="xs:int" default="0"
                use="optional" form="qualified"/>
  <xs:attribute name="HeaderRowDelimiter" type="xs:string" use="optional"
                form="qualified"/>
  <xs:attribute name="ColumnNamesInFirstDataRow" default="False"
                type="DTS:BooleanStringCap" use="optional" form="qualified"/>
  <xs:attribute name="RowDelimiter" type="xs:string" use="optional"
                form="qualified"/>
  <xs:attribute name="DataRowsToSkip" type="xs:int" default="0"
                use="optional" form="qualified"/>
  <xs:attribute name="TextQualifier" type="xs:string" use="optional"
                form="qualified"/>
  <xs:attribute name="CodePage" type="xs:int" use="optional"
                form="qualified"/>
  <xs:attribute name="ServerName" type="xs:string" use="optional"
```

```
                    form="qualified"/>
    <xs:attribute name="UseFile" type="DTS:BooleanStringCap"
                    default="False" use="optional" form="qualified"/>
    <xs:attribute name="UseEncryption" type="DTS:BooleanStringCap"
                    default="False" use="optional" form="qualified"/>
    <xs:attribute name="RetainData" type="DTS:BooleanStringCap"
                    default="True" use="optional" form="qualified"/>
</xs:attributeGroup>
```

The following table specifies the attributes for the **ConnectionManagerConnectionManagerAttributeGroup** attribute group.

| Attribute | Description |
|---|---|
| Retain | A value of type **DTS:BooleanStringCap** that specifies whether the same connection is retained between successive data accesses. False specifies that the connection is not retained between successive data accesses. True specifies that the connection is retained between successive data accesses. |
| ConnectionString | A string value that is handed to the data source upon connection. The contents of the string vary with the source type. |
| FileUsageType | An enumeration value that specifies how the file connection is used. |
| Format | An enumeration value that specifies the type of formatting in the records and fields of a file. |
| LocaleID | An integer value that specifies the language code identifier (LCID) for the language for this executable.<7> |
| Unicode | A value of type **DTS:BooleanStringCap** that specifies an indicator for whether the data source contains Unicode characters. False indicates that the data obtained from the connection is not Unicode. True specifies that the data obtained from the connection is Unicode. |
| HeaderRowsToSkip | An integer value that represents the number of header rows to skip in the data source. |
| HeaderRowDelimiter | A string value that specifies a hexadecimal-encoded value in ASCII [US-ASCII] for the character or character string that is the delimiter for the header row. Each such character MUST be prepended and post-pended with an underscore (_) character. For example, the string "_x000D__x000A_" specifies the sequence of characters: a carriage return character followed by a line feed character. |
| ColumnNamesInFirstDataRow | An enumeration value that specifies whether the first row contains column names. False specifies that the first row does not contain column names. True specifies that the first data row contains column names. |
| RowDelimiter | A string value that specifies the row delimiter.<8> |
| DataRowsToSkip | An integer value that specifies the number of data rows to skip. |
| TextQualifier | A string value that specifies a character that is used to indicate text values. |
| CodePage | Specifies that the element value is the numeric value of the ANSI code page that is used for strings.<9> |
| ServerName | A string value that specifies the name of the server for the connection. |
| UseFile | A value of type **DTS:BooleanStringCap** that specifies whether the cache is |

| Attribute | Description |
|---|---|
| | backed by a cache file on disk or stored only in memory. |
| | True specifies that a file is used. |
| | False specifies that the cache is stored only in memory and erased when the package terminates. |
| UseEncryption | A value of type **DTS:BooleanStringCap** that specifies whether to use encryption for data access. |
| | True specifies to use encryption for data access. |
| | False specifies not to use encryption for data access. |
| RetainData | A value of type **DTS:BooleanStringCap** that specifies whether data is retained.<10> |
| | True specifies that data is retained. |
| | False specifies that data is not retained. |

#### 2.4.4.1.2.1.1.1    FormatEnum

The **FormatEnum** type contains the enumeration values that specify how the file connection is used.

The following is the XSD of the **FormatEnum** type.

```
<xs:simpleType name="FormatEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Delimited"/>
    <xs:enumeration value="FixedWidth"/>
    <xs:enumeration value="RaggedRight"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the descriptions for the enumeration values of the **FormatEnum** type.

| Element value | Description |
|---|---|
| Delimited | Specifies that the format of the file contains fields that are delimited with a delimiter. |
| FixedWidth | Specifies that the format of the file contains fields that are of fixed width. |
| RaggedRight | Specifies that the format of the file contains records that can be truncated on the right. |

#### 2.4.4.1.2.1.1.2    FileUsageTypeEnum

The **FileUsageTypeEnum** type contains the enumeration values that specify how the file connection is used.

The following is the XSD of the **FileUsageTypeEnum** type.

```
<xs:simpleType name="FileUsageTypeEnum">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="3"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the descriptions for the enumeration values of the **FileUsageTypeEnum** type.

| Enumeration value | Description |
|---|---|
| 0 | Specifies that the file connection manager uses an existing file. |
| 1 | Specifies that the file connection manager creates a file. |
| 2 | Specifies that the file connection manager uses an existing folder. |
| 3 | Specifies that the file connection manager creates a folder. |

### 2.4.4.1.2.1.2 FlatFileColumnsType

The **FlatFileColumnsType** complex type is the container type for a collection of elements of type FlatFileColumnType.

The following is the XSD of the **FlatFileColumnsType** complex type.

```
<xs:complexType name="FlatFileColumnsType">
  <xs:sequence>
    <xs:element name="FlatFileColumn" type="DTS:FlatFileColumnType" minOccurs="0"
 maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

The following table provides additional information about the elements, types, and constraints for the **FlatFileColumnsType** complex type.

| Element | Type definition | Description |
|---|---|---|
| FlatFileColumn | FlatFileColumnType | Specifies a **FlatFileColumn** element that contains the options, parameters, and settings for a column of a flat file. |

### 2.4.4.1.2.1.2.1 FlatFileColumnType

The **FlatFileColumnType** complex type gives the definition of a column that is contained in a flat file data source.

The following is the XSD of the **FlatFileColumnType** complex type.

```
<xs:complexType name="FlatFileColumnType">
  <xs:sequence/>
  <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
  <xs:attributeGroup ref="DTS:FlatFileColumnAttributeGroup"/>
</xs:complexType>
```

The **FlatFileColumnType** complex type contains no elements.

The following table provides additional information about the attributes, attribute groups, and types for the **ConfigurationType** complex type.

| Attribute | Description |
|---|---|
| BasePropertyAttributeGroup | An attribute group that specifies attributes that are common to many different types throughout the schema. |
| FlatFileColumnAttributeGroup | An attribute group that specifies attributes for the **FlatFileColumnType** complex type. |

### 2.4.4.1.2.1.2.1.1    FlatFileColumnAttributeGroup

The **FlatFileColumnAttributeGroup** attribute group contains the attributes for the FlatFileColumnType type.

The following is the XSD for the **FlatFileColumnAttributeGroup** attribute group.

```
<xs:attributeGroup name="FlatFileColumnAttributeGroup">
  <xs:attribute name="ColumnType" type="DTS:ColumnTypeEnum"
               default="FixedWidth" use="optional" form="qualified"/>
  <xs:attribute name="ColumnDelimiter" type="xs:string"
               use="required" form="qualified"/>
  <xs:attribute name="ColumnWidth" type="xs:int" default="0"
               use="optional" form="qualified"/>
  <xs:attribute name="MaximumWidth" type="xs:int" default="0"
               use="optional" form="qualified"/>
  <xs:attribute name="DataType" type="DTS:DtsDataTypeEnum" default="0"
               use="optional" form="qualified"/>
  <xs:attribute name="DataPrecision" type="xs:int" default="0"
               use="optional" form="qualified"/>
  <xs:attribute name="DataScale" type="xs:int" default="0" use="optional"
               form="qualified"/>
  <xs:attribute name="TextQualified" type="DTS:BooleanStringCap"
               default="False" use="optional" form="qualified"/>
</xs:attributeGroup>
```

The following table specifies the attributes for the **FlatFileColumnAttributeGroup** attribute group.

| Attribute | Description |
|---|---|
| ColumnType | An enumeration value that specifies whether the column is delimited or fixed width. |
| ColumnDelimiter | A string value that specifies the column delimiter in the file. |
| ColumnWidth | An integer value that specifies the width of the column. This value is used if the **ColumnType** property value is "FixedWidth". |
| MaximumWidth | An integer value that specifies the maximum number of bytes to be read from the buffer. |
| DataType | An enumeration value that specifies the data type of the column. |
| DataPrecision | An integer value that specifies the total digits in a numeric data type. |
| DataScale | An integer value that specifies the number of digits to the right of the decimal point. |
| TextQualified | A value of type **DTS:BooleanStringCap** that specifies whether the column is text-qualified. "True" specifies that the column is text-qualified. "False" specifies that the column is not text-qualified. |

#### 2.4.4.1.2.1.2.1.2   ColumnTypeEnum

The **ColumnTypeEnum** type contains the enumeration values that specify whether the column is delimited or of fixed width.

The following is the XSD of the **ColumnTypeEnum** type.

```
<xs:simpleType name="ColumnTypeEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Delimited"/>
    <xs:enumeration value="FixedWidth"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the descriptions for the enumeration values of the **ColumnTypeEnum** type.

| Enumeration value | Description |
|---|---|
| Delimited | Specifies that the column values are delimited by a delimiter. |
| FixedWidth | Specifies that the column values are of fixed width. |

#### 2.4.4.1.2.1.3 CacheColumnsType

The **CacheColumnsType** complex type is the container type for a collection of elements of type CacheColumnType.

The following is the XSD of the **CacheColumnsType** type.

```
<xs:complexType name="CacheColumnsType">
  <xs:sequence>
    <xs:element name="CacheColumn" type="DTS:CacheColumnType"
                minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

The following table provides additional information about the elements, types, and constraints for the **CacheColumnsType** complex type.

| Element | Type definition | Description |
|---|---|---|
| CacheColumn | CacheColumnType | Specifies a **CacheColumn** element that contains the options, parameters, and settings for a cache column. |

#### 2.4.4.1.2.1.3.1   CacheColumnType Complex Type

The **CacheColumnType** complex type specifies the definition of a column that is contained in a Cache data source.

The following is the XSD for the **CacheColumnType** complex type.

```
<xs:complexType name="CacheColumnType">
  <xs:sequence/>
```

```
        <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
        <xs:attributeGroup ref="DTS:CacheColumnAttributeGroup"/>
    </xs:complexType>
```

The **CacheColumnType** complex type contains no elements.

The following table provides additional information about the attributes, attribute groups, and types for the **CacheColumnType** complex type.

| Attribute | Description |
|---|---|
| BasePropertyAttributeGroup | An attribute group that specifies attributes that are common to many different types throughout the schema. |
| CacheColumnAttributeGroup | An attribute group that specifies attributes for the **CacheColumnType** complex type. |

### 2.4.4.1.2.1.3.1.1    CacheColumnAttributeGroup

The **CacheColumnAttributeGroup** attribute group contains the attribute for the CacheColumnType complex type.

The following is the XSD for the **CacheColumnAttributeGroup** attribute group.

```
    <xs:attributeGroup name="CacheColumnAttributeGroup">
      <xs:attribute name="DataType" type="DTS:DtsDataTypeEnum" default="0"
                    use="optional" form="qualified"/>
      <xs:attribute name="Length" type="xs:int" default="0" use="optional"
                    form="qualified"/>
      <xs:attribute name="Precision" type="xs:int" default="0" use="optional"
                    form="qualified"/>
      <xs:attribute name="Scale" type="xs:int" default="0" use="optional"
                    form="qualified"/>
      <xs:attribute name="CodePage" type="xs:int" default="0" use="optional"
                    form="qualified"/>
      <xs:attribute name="IndexPosition" type="xs:int" default="0"
                    use="optional" form="qualified"/>
    </xs:attributeGroup>
```

The following table specifies the attributes for the **CacheColumnAttributeGroup** attribute group.

| Attribute | Description |
|---|---|
| DataType | An enumeration value that specifies the data type of the column. |
| Length | An integer value that specifies the length of the cache column, in characters. |
| Precision | An integer value that specifies an integer that defines the total digits in a numeric data type. |
| Scale | An integer value that specifies the number of digits to the right of the decimal point. |
| CodePage | An integer value that specifies that the element value is the numeric value of the ANSI code page that is used for strings. |
| IndexPosition | An integer value that specifies the index position of the current cache column. |

### 2.4.4.1.2.1.4 FtpConnectionType Complex Type

The **FtpConnectionType** complex type specifies the properties that are necessary to connect to an FTP data source.

The following is the XSD of the **FtpConnectionType** complex type.

```
<xs:complexType name="FtpConnectionType">
  <xs:sequence>
    <xs:element name="Property"
                minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:complexContent>
          <xs:extension base="DTS:PropertyElementBaseType">
            <xs:attribute name="Name"
                type="DTS:FtpConnectionPropertyNameEnum"
                use="required"/>
            <xs:attribute name="Sensitive" form="unqualified">
              <xs:simpleType>
                <xs:restriction base="xs:int">
                  <xs:minInclusive value="0"/>
                  <xs:maxInclusive value="1"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="Encrypted" form="unqualified">
              <xs:simpleType>
                <xs:restriction base="xs:int">
                  <xs:minInclusive value="0"/>
                  <xs:maxInclusive value="1"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:attribute>
          </xs:extension>
        </xs:complexContent>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attributeGroup ref="DTS:FtpConnectionAttributeGroup"/>
</xs:complexType>
```

The following table provides additional information about the elements, types, and constraints in the **FtpConnectionType** complex type.

| Element | Additional constraints | Type definition | Description |
|---------|------------------------|-----------------|-------------|
| Property | In some cases, constraints are placed on the **Property** element value for specific values of the **Name** attribute of the **Property** element. If such constraints exist, they are specified where the meaning of the particular property name is specified. These constraints MUST be followed. If the optional **DataType** attribute is present, the value of the **Property** element MUST be of the type that is specified by the **DataType** attribute. | Inline in the XSD in this section. | **Property** elements contain values for many different properties associated with an instance of the containing **FtpConnection** element, according to the **Name** attribute of each **Property** element. |

The following table provides additional information about the attributes and types for the **Property** element that is declared inline in the complex type declaration. The **Property** element is an extension of the PropertyElementBaseType type, with the addition of the attributes that are specified in the following table.

| Attribute | Constraints | Description |
|-----------|-------------|-------------|
| Name | None | Specifies the name of the **Property** element. Each named property has its own use and interpretation. The explanation for the interpretation of each named property is given with the definition of the type for the **Name** attribute. |
| Sensitive | None | The element value contains an indicator for whether the property information is treated as sensitive. The integer value has the following meanings:<br><br>0 specifies that the property value is not considered to be sensitive.<br><br>1 specifies that the property value is considered to be sensitive.<br><br>Depending on the setting of the **ProtectionLevel** value in an ancestor level to the containing element, the value of the property can or cannot be displayed if the property is considered to be sensitive. |
| Encrypted | None | The element value contains an indicator for whether the property element value is encrypted. The integer value has the following meanings:<br><br>0 specifies that the property value is not encrypted.<br><br>1 specifies that the property value is encrypted. |

#### 2.4.4.1.2.1.4.1 FtpConnectionPropertyNameEnum

The **FtpConnectionPropertyNameEnum** type contains the values for the **Name** attribute of the property elements that are allowed for the **FtpConnectionType** type.

The following is the XSD for the **FtpConnectionPropertyNameEnum** type.

```
<xs:simpleType name="FtpConnectionPropertyNameEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="ServerPassword"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **FtpConnectionPropertyNameEnum** type.

| Enumeration value | Description |
|-------------------|-------------|
| ServerPassword | Specifies a string that contains the password to use to access the FTP server. |

#### 2.4.4.1.2.1.4.2 FtpConnectionAttributeGroup

The **FtpConnectionAttributeGroup** attribute group contains the attribute for the FtpConnectionType complex type.

The following is the XSD for the **FtpConnectionAttributeGroup** attribute group.

```
<xs:attributeGroup name="FtpConnectionAttributeGroup">
  <xs:attribute name="ServerName" type="xs:string" use="required"
                form="qualified"/>
  <xs:attribute name="ServerPort" type="xs:int" default="21"
                use="optional" form="qualified"/>
  <xs:attribute name="ServerUserName" type="xs:string" default=""
                use="optional" form="qualified"/>
  <xs:attribute name="ChunkSize" type="xs:int" default="1" use="optional"
```

```
                         form="qualified"/>
    <xs:attribute name="TimeOut" type="xs:int" default="60" use="optional"
                         form="qualified"/>
    <xs:attribute name="Retries" type="xs:int" default="5" use="optional"
                         form="qualified"/>
    <xs:attribute name="UsePassiveMode" type="DTS:BooleanStringCap"
                         default="False" use="optional" form="qualified"/>
  </xs:attributeGroup>
```

The following table specifies the attributes for the **FtpConnectionAttributeGroup** attribute group.

| Enumeration value | Description |
|---|---|
| ServerName | A string value that specifies the name of the FTP server to connect to. |
| ServerPort | An integer value that specifies the port for the connection. |
| ServerUserName | A string value that specifies the user name to use on the FTP server. |
| ChunkSize | An integer value that specifies the minimum number of bytes to receive or send. |
| TimeOut | An integer value that specifies the number of seconds to wait before the connection fails. |
| Retries | An integer value that specifies the number of times that the FTP connection attempts to send data before the connection fails. |
| UsePassiveMode | A value of type **DTS:BooleanStringCap** that specifies that the element value contains an indicator for whether the FTP connection uses passive mode. "False" specifies that active mode is used for the FTP connection. "True" specifies that passive mode is used for the FTP connection. |

### 2.4.4.1.2.1.5 HttpConnectionType Complex Type

The **HttpConnectionType** complex type specifies the properties that are necessary to connect to an HTTP data source.

The following is the XSD of the **HttpConnectionType** complex type.

```
    <xs:complexType name="HttpConnectionType">
      <xs:sequence>
        <xs:element name="Property" minOccurs="0" maxOccurs="unbounded">
          <xs:complexType>
            <xs:complexContent>
              <xs:extension base="DTS:PropertyElementBaseType">
                <xs:attribute name="Name" type="DTS:HttpConnectionPropertyNameEnum"
                     use="required"/>
                <xs:attribute name="Sensitive" form="unqualified">
                  <xs:simpleType>
                    <xs:restriction base="xs:int">
                      <xs:minInclusive value="0"/>
                      <xs:maxInclusive value="1"/>
                    </xs:restriction>
                  </xs:simpleType>
                </xs:attribute>
                <xs:attribute name="Encrypted" form="unqualified">
                  <xs:simpleType>
                    <xs:restriction base="xs:int">
                      <xs:minInclusive value="0"/>
                      <xs:maxInclusive value="1"/>
                    </xs:restriction>
```

```
          </xs:simpleType>
        </xs:attribute>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
</xs:sequence>
<xs:attributeGroup ref="DTS:HttpConnectionAttributeGroup"/>
</xs:complexType>
```

The following table provides additional information about the attributes and types for the **Property** element that is declared inline in the complex type declaration. The **Property** element is an extension of the PropertyElementBaseType type, with the addition of the attributes that are specified in the following table.

| Attribute | Constraints | Description |
|-----------|-------------|-------------|
| Name | None | Specifies the name of the **Property** element. Each named property has its own use and interpretation. The explanation for the interpretation of each named property is given with the definition of the type for the **Name** attribute. |
| Sensitive | None | The element value contains an indicator for whether the property information is treated as sensitive. The integer value has the following meanings:<br><br>0 specifies that the property value is not considered to be sensitive.<br><br>1 specifies that the property value is considered to be sensitive.<br><br>Depending on the setting of the **ProtectionLevel** value in an ancestor level to the containing element, the value of the property can or cannot be displayed if the property is considered to be sensitive. |
| Encrypted | None | The element value contains an indicator for whether the property element value is encrypted. The integer value has the following meanings:<br><br>0 specifies that the property value is not encrypted.<br><br>1 specifies that the property value is encrypted. |

#### 2.4.4.1.2.1.5.1 HttpConnectionPropertyNameEnum

The **HttpConnectionPropertyNameEnum** type contains the values for the **Name** attribute of the **Property** elements that are allowed for the HttpConnectionType complex type.

The following is the XSD for the **HttpConnectionPropertyNameEnum** type.

```
<xs:simpleType name="HttpConnectionPropertyNameEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="ProxyPassword"/>
    <xs:enumeration value="ServerPassword"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **HttpConnectionPropertyNameEnum** type.

| Enumeration value | Description |
|-------------------|-------------|
| ServerPassword | Specifies a string value that contains the password to use on the server. |
| ProxyPassword | Specifies a string value that contains the password for the **ProxyUserName** account on |

| Enumeration value | Description |
|---|---|
| | the proxy server. |

### 2.4.4.1.2.1.5.2 HttpConnectionAttributeGroup

The **HttpConnectionAttributeGroup** attribute group contains the attribute for the HttpConnectionType complex type.

The following is the XSD for the **HttpConnectionAttributeGroup** attribute group.

```
<xs:attributeGroup name="HttpConnectionAttributeGroup">
   <xs:attribute name="ServerURL" type="xs:string" use="required"
                 form="qualified"/>
   <xs:attribute name="UseServerCredentials" type="DTS:BooleanStringCap"
                 default="False" use="optional" form="qualified"/>
   <xs:attribute name="ServerUserName" type="xs:string" default=""
                 use="optional" form="qualified"/>
   <xs:attribute name="ServerDomain" type="xs:string" default=""
                 use="optional" form="qualified"/>
   <xs:attribute name="UseSecureConnection" type="DTS:BooleanStringCap"
                 default="False" use="optional" form="qualified"/>
   <xs:attribute name="CertificateIdentifier" type="xs:string" default=""
                 use="optional" form="qualified"/>
   <xs:attribute name="Certificate" type="xs:string" default=""
                 use="optional" form="qualified"/>
   <xs:attribute name="TimeOut" type="xs:int" default="60" use="optional"
                 form="qualified"/>
   <xs:attribute name="ChunkSize" type="xs:int" default="1" use="optional"
                 form="qualified"/>
   <xs:attribute name="UseProxy" type="DTS:BooleanStringCap"
                 default="False" use="optional" form="qualified"/>
   <xs:attribute name="ProxyURL" type="xs:string" default="" use="optional"
                 form="qualified"/>
   <xs:attribute name="BypassProxyOnLocal" type="DTS:BooleanStringCap"
                 default="False" use="optional" form="qualified"/>
   <xs:attribute name="UseProxyCredentials" type="DTS:BooleanStringCap"
                 default="False" use="optional" form="qualified"/>
   <xs:attribute name="ProxyUserName" type="xs:string" default=""
                 use="optional" form="qualified"/>
   <xs:attribute name="ProxyDomain" type="xs:string" default=""
                 use="optional" form="qualified"/>
   <xs:attribute name="BypassList" type="xs:string" default=""
                 use="optional" form="qualified"/>
</xs:attributeGroup>
```

The following table specifies the attributes for the **HttpConnectionAttributeGroup** attribute group.

| Enumeration value | Description |
|---|---|
| ServerURL | A string value that specifies the URL location of the HTTP server. |
| UseServerCredentials | A value of type **DTS:BooleanStringCap** that specifies an indicator for whether to use network credentials.<br>"True" specifies that the connection uses server credentials.<br>"False" specifies that the connection does not use server credentials. |
| ServerUserName | A string value that specifies the user name to use to log on to the server. |
| ServerDomain | A string value that specifies the domain name for the server. |

| Enumeration value | Description |
|---|---|
| UseSecureConnection | A value of type **DTS:BooleanStringCap** that specifies whether to use an HTTP or an HTTPS connection.<br><br>"False" specifies to use the HTTP protocol.<br><br>"True" specifies to use the HTTPS protocol. |
| CertificateIdentifier | A string value that specifies the X.509 [RFC2459] certificate identifier. |
| Certificate | A string that specifies the client certificate to the HTTP server in base64. |
| TimeOut | An integer value that specifies the number of seconds after which the connection times out. |
| ChunkSize | An integer value that specifies the minimum number of bytes to read or write from the server. |
| UseProxy | A value of type **DTS:BooleanStringCap** that specifies whether to use a proxy server.<br><br>"True" specifies that the connection uses a proxy server.<br><br>"False" specifies that the connection does not use a proxy server. |
| ProxyURL | A string value that specifies the URL location of the proxy server. |
| BypassProxyOnLocal | A value of type **DTS:BooleanStringCap** that specifies whether to bypass a proxy server and use a local address instead.<br><br>"True" specifies that the connection bypasses the proxy server for local addresses.<br><br>"False" specifies that the connection does not bypass the proxy server for local addresses. |
| UseProxyCredentials | A value of type **DTS:BooleanStringCap** that specifies whether to use proxy credentials.<br><br>"True" specifies that the connection uses proxy credentials.<br><br>"False" specifies that the connection does not use proxy credentials. |
| ProxyUserName | A string value that specifies the name of the user to use to log on to the proxy server. |
| ProxyDomain | A string value that specifies the domain name for the proxy server. |
| BypassList | A string value that specifies a list of addresses that do not use the proxy server. |

#### 2.4.4.1.2.2 ConnectionManagerObjectDataMsmqConnectionManagerType

The **ConnectionManagerObjectDataMsmqConnectionManagerType** complex type is the type of the **MsmqConnectionManager** element that is a child of the **ObjectData** element when the type of the connection is to an MSMQ source.

The following is the XSD of the **ConnectionManagerObjectDataMsmqConnectionManagerType** complex type.

```
<xs:complexType
  name="ConnectionManagerObjectDataMsmqConnectionManagerType">
  <xs:sequence/>
  <xs:attribute name="ConnectionString" type="xs:string" form="unqualified"/>
</xs:complexType>
```

The **ConnectionManagerObjectDataMsmqConnectionManagerType** complex type MUST NOT contain any elements.

The following table specifies the attributes for the
**ConnectionManagerObjectDataMsmqConnectionManagerType** complex type.

| Attribute | Constraints | Description |
|-----------|-------------|-------------|
| ConnectionString | None | The connection string to an MSMQ source. |

### 2.4.4.1.2.3 ConnectionManagerObjectDataSMOServerConnectionManagerType

The **ConnectionManagerObjectDataSMOServerConnectionManagerType** complex type is the
type of the **SMOServerConnectionManager** element that is a child of the **ObjectData** element
when the type of the connection is to a database that is using the Server Management Objects (SMO)
object model for server management.<11>

The following is the XSD of the
**ConnectionManagerObjectDataSMOServerConnectionManagerType** complex type.

```
<xs:complexType
  name="ConnectionManagerObjectDataSMOServerConnectionManagerType">
  <xs:sequence>
    <xs:element name="PassWord" type= "DTS:PasswordElementType" form="unqualified"/>
  </xs:sequence>
  <xs:attribute name="ConnectionString" type="xs:string" form="unqualified"
use="required"/>
</xs:complexType>
```

The following tables specify the elements and attributes for the
**ConnectionManagerObjectDataSMOServerConnectionManagerType** complex type.

| Element | Constraints | Description |
|---------|-------------|-------------|
| PassWord | None | The password for the connection to the SMO server source. The password can be encrypted, depending on the settings of the attribute values for the **PassWord** element. |

| Attribute | Constraints | Description |
|-----------|-------------|-------------|
| ConnectionString | None | The connection string to an SMO server source. |

### 2.4.4.1.2.4 ConnectionManagerObjectDataSmtpConnectionManagerType

The **ConnectionManagerObjectDataSmtpConnectionManagerType** complex type is the type of
the **SmtpConnectionManager** element, which is a child of the **ObjectData** element when the type
of the connection is to an SMTP server.

The following is the XSD of the **ConnectionManagerObjectDataSmtpConnectionManagerType**
complex type.

```
<xs:complexType name="ConnectionManagerObjectDataSmtpConnectionManagerType">
  <xs:sequence/>
  <xs:attribute name="ConnectionString" type="xs:string" form="unqualified"/>
```

```
        </xs:complexType>
```

The **ConnectionManagerObjectDataSmtpConnectionManagerType** complex type MUST NOT contain any elements.

The following table specifies the attributes for the **ConnectionManagerObjectDataSmtpConnectionManagerType** complex type.

| Attribute | Constraints | Description |
|---|---|---|
| ConnectionString | None | The connection string to an SMTP source. |

### 2.4.4.1.2.5   ConnectionManagerObjectDataWmiConnectionManagerType

The **ConnectionManagerObjectDataWmiConnectionManagerType** complex type is the type of the **WmiConnectionManager** element, which is a child of the **ObjectData** element when the type of the connection is to a WMI data source.

The following is the XSD of the **ConnectionManagerObjectDataWmiConnectionManagerType** complex type.

```
    <xs:complexType name="ConnectionManagerObjectDataWmiConnectionManagerType">
      <xs:sequence>
        <xs:element name="PassWord" minOccurs="0" type="DTS:PasswordElementType"
 form="unqualified"/>
      </xs:sequence>
      <xs:attribute name="ConnectionString" type="xs:string" form="unqualified"/>
    </xs:complexType>
```

The following tables specify the elements and attributes for the **ConnectionManagerObjectDataWmiConnectionManagerType** complex type.

| Element | Constraints | Description |
|---|---|---|
| PassWord | None | The password for the connection to a WMI data source. The password can be encrypted, depending on the settings of the attribute values for the **PassWord** element. |

| Attribute | Constraints | Description |
|---|---|---|
| ConnectionString | None | Specifies that the connection string is to a WMI data source. |

### 2.4.4.1.3 ConnectionManagerObjectDataType Information for Data Sources

The type of data source that the connection manager specifies access for carries with it important constraints that are not fully expressed in the XSD code fragments. These additional constraints, which are explained in the subsections of this section, MUST be followed for each specific data source type. The important differences that vary by the data source type are the following:

- Allowed values for the **Name** attribute of the **Property** element within the **ObjectData** element can differ between different data sources. The same limitations apply to the **PropertyExpression** element within a **ConnectionManager** instance.

- The choice contained within the complex type definition for ConnectionManagerObjectDataType is also constrained by the type of data source. Not all of the choices in the XSD are available for each data source type.

- For those data sources that allow a **ConnectionManager** element to be contained by the **ObjectData** element, the choice that is contained within the complex type definition for the **ConnectionManager** element of the **ConnectionManagerObjectDataType** is also constrained by the type of data source. Not all of the choices in the XSD are available for each data source type.

### 2.4.4.1.3.1 OLE DB Data Source

When the connection manager is to an OLE DB data source, the formal type of the **ObjectData** element is the ConnectionManagerObjectDataType type. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type from the restrictions that are declared in the formal type declaration. A **ConnectionManager** instance for an OLE DB source MUST follow the restrictions that are shown in this anonymous XSD fragment.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **ConnectionManagerObjectDataType** type:

- The value for the **CreationName** attribute of the containing **ConnectionManager** element MUST be "OLEDB".

- The **ObjectData** element type MUST be of the type that is specified by the anonymous XSD definition that is contained in this section.

- The **ObjectData** element MUST contain a **ConnectionManager** element.

- The **ConnectionManager** child element to the **ObjectData** element MUST NOT include attributes, except for those that are specified as valid in this anonymous XSD fragment, which is a subset of the attributes that are contained in the ConnectionManagerConnectionManagerAttributeGroup attribute group. The **PropertyExpression** element of the containing **ConnectionManager** element MUST NOT include properties, except for those that are named the same as the attributes for the **ConnectionManager** element in this anonymous type definition XSD fragment.

This anonymous type declaration acts as a replacement for the definition for **ConnectionManagerObjectDataType** type in the case of an OLE DB data source.

```
  <xs:complexType>
    <xs:choice>
      <xs:element name="ConnectionManager">
        <xs:complexType>
          <xs:sequence/>
          <xs:attribute name="Retain" type="DTS:BooleanStringCap" default="False"
  use="optional" form="qualified"/>
          <xs:attribute name="ConnectionString" type="xs:string" use="required"
  form="qualified"/>
        </xs:complexType>
      </xs:element>
    </xs:choice>
  </xs:complexType>
```

### 2.4.4.1.3.2 Flat File Data Source

When the connection manager is to a flat file data source, the formal type of the **ObjectData** element is the ConnectionManagerObjectDataType type. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type from the restrictions that are declared in the formal type declaration. A **ConnectionManager** instance for a flat file data source MUST follow the restrictions that are shown in this anonymous XSD fragment.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **ConnectionManagerObjectDataType** type:

- The value for the **CreationName** attribute of the containing **ConnectionManager** element MUST be "FLATFILE".

- The **ObjectData** element type MUST be of the type that is specified by the anonymous XSD definition that is contained in this section.

- The **ObjectData** element MUST contain a **ConnectionManager** element.

- The **ConnectionManager** child element to the **ObjectData** element MUST NOT include attributes, except for those that are specified as valid in this anonymous XSD fragment, which is a subset of the attributes that are contained in the ConnectionManagerConnectionManagerAttributeGroup attribute group. The **PropertyExpression** element of the containing **ConnectionManager** element MUST NOT include properties, except for those that are named the same as the attributes for the **ConnectionManager** element in this anonymous type definition XSD fragment.

- The following attributes are required for a flat file data source and MUST be specified: **Format**, **HeaderRowDelimiter**, **RowDelimiter**, and **TextQualifier**.

This anonymous type declaration acts as a replacement for the definition for **ConnectionManagerObjectDataType** type in the case of a flat file data source.

```
<xs:complexType>
  <xs:choice>
    <xs:element name="ConnectionManager">
      <xs:complexType>
        <xs:sequence/>
        <xs:attribute name="ConnectionString" type="xs:string"
                      use="required" form="qualified"/>
        <xs:attribute name="FileUsageType" type="DTS:FileUsageTypeEnum"
                      default="0" use="optional" form="qualified"/>
        <xs:attribute name="Format" type="DTS:FormatEnum" use="required"
                      form="qualified"/>
        <xs:attribute name="LocaleID" type="xs:int" use="optional"
                      form="qualified"/>
        <xs:attribute name="Unicode" type="DTS:BooleanStringCap"
                      default="False" use="optional" form="qualified"/>
        <xs:attribute name="HeaderRowsToSkip" type="xs:int" default="0"
                      use="optional" form="qualified"/>
        <xs:attribute name="HeaderRowDelimiter" type="xs:string"
                      use="required" form="qualified"/>
        <xs:attribute name="ColumnNamesInFirstDataRow" default="False"
                      type="DTS:BooleanStringCap" use="optional"
                      form="qualified"/>
        <xs:attribute name="RowDelimiter" type="xs:string" use="required"
                      form="qualified"/>
        <xs:attribute name="DataRowsToSkip" type="xs:int" default="0"
                      use="optional" form="qualified"/>
        <xs:attribute name="TextQualifier" type="xs:string" use="required"
                      form="qualified"/>
        <xs:attribute name="CodePage" type="xs:int" use="optional"
                      form="qualified"/>
      </xs:complexType>
    </xs:element>
  </xs:choice>
```

```
        </xs:complexType>
```

### 2.4.4.1.3.3   ADO.NET Data Source

When the connection manager is to an ADO.NET data source, the formal type of the **ObjectData** element is the ConnectionManagerObjectDataType type. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type from the restrictions that are declared in the formal type declaration. A **ConnectionManager** instance for an ADO.NET source MUST follow the restrictions that are shown in this anonymous XSD fragment.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **ConnectionManagerObjectDataType** type:

▪   The value for the **CreationName** attribute of the containing **ConnectionManager** element MUST be "ADO.NET" concatenated together with the fully qualified name of a valid ADO.NET assembly.

▪   The **ObjectData** element type MUST be of the type that is specified by the anonymous XSD definition that is contained in this section.

▪   The **ObjectData** element MUST contain a **ConnectionManager** element.

▪   The **ConnectionManager** child element to the **ObjectData** element MUST NOT include attributes, except for those that are specified as valid in this anonymous XSD fragment, which is a subset of the attributes that are contained in the ConnectionManagerConnectionManagerAttributeGroup attribute group. The **PropertyExpression** element of the containing **ConnectionManager** element MUST NOT include properties, except for those that are named the same as the attributes for the **ConnectionManager** element in this anonymous type definition XSD fragment.

This anonymous type declaration acts as a replacement for the definition for **ConnectionManagerObjectDataType** type in the case of an ADO.NET data source.

```
<xs:complexType>
  <xs:choice>
    <xs:element name="ConnectionManager">
      <xs:complexType>
        <xs:sequence/>
        <xs:attribute name="Retain" type="DTS:BooleanStringCap"
                      default="False" use="optional" form="qualified"/>
        <xs:attribute name="ConnectionString" type="xs:string"
                      use="required" form="qualified"/>
        <xs:attribute name="ServerName" type="xs:string" use="required"
                      form="qualified"/>
      </xs:complexType>
    </xs:element>
  </xs:choice>
</xs:complexType>
```

### 2.4.4.1.3.4   Analysis Services Data Source

When the connection manager is to an Analysis Services data source, the formal type of the **ObjectData** element is the ConnectionManagerObjectDataType type. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type from the restrictions that are declared in the formal type declaration. A **ConnectionManager** instance for an Analysis Services source MUST follow the restrictions that are shown in this anonymous XSD fragment.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **ConnectionManagerObjectDataType** type:

- The value for the **CreationName** attribute of the containing **ConnectionManager** element MUST be "MSOLAP100".

- The **ObjectData** element type MUST be of the type that is specified by the anonymous XSD definition that is contained in this section.

- The **ObjectData** element MUST contain a **ConnectionManager** element.

- The **ConnectionManager** child element to the **ObjectData** element MUST NOT include attributes, except for those that are specified as valid in this anonymous XSD fragment, which is a subset of the attributes that are contained in the ConnectionManagerConnectionManagerAttributeGroup attribute group. The **PropertyExpression** element of the containing **ConnectionManager** element MUST NOT include properties, except for those that are named the same as the attributes for the **ConnectionManager** element in this anonymous type definition XSD fragment.

This anonymous type declaration acts as a replacement for the definition for **ConnectionManagerObjectDataType** type in the case of an Analysis Services data source.

```
<xs:complexType>
  <xs:choice>
    <xs:element name="ConnectionManager">
      <xs:complexType>
        <xs:sequence/>
        <xs:attribute name="Retain" type="DTS:BooleanStringCap"
                      default="False" use="optional" form="qualified"/>
        <xs:attribute name="ConnectionString" type="xs:string"
                      use="required" form="qualified"/>
      </xs:complexType>
    </xs:element>
  </xs:choice>
</xs:complexType>
```

### 2.4.4.1.3.5   File Data Source

When the connection manager is to a File data source, the formal type of the **ObjectData** element is the ConnectionManagerObjectDataType type. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type from the restrictions that are declared in the formal type declaration. A **ConnectionManager** instance for a File source MUST follow the restrictions that are shown in this anonymous XSD fragment.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **ConnectionManagerObjectDataType** type:

- The value for the **CreationName** attribute of the containing **ConnectionManager** element MUST be "FILE".

- The **ObjectData** element type MUST be of the type that is specified by the anonymous XSD definition that is contained in this section.

- The **ObjectData** element MUST contain a **ConnectionManager** element.

- The **ConnectionManager** child element to the **ObjectData** element MUST NOT include attributes, except for those that are specified as valid in this anonymous XSD fragment, which is a subset of the attributes that are contained in the ConnectionManagerConnectionManagerAttributeGroup attribute group. The **PropertyExpression** element of the containing **ConnectionManager** element MUST NOT include properties, except for those that are named the same as the attributes for the **ConnectionManager** element in this anonymous type definition XSD fragment.

This anonymous type declaration acts as a replacement for the definition for **ConnectionManagerObjectDataType** type in the case of a File data source.

```
<xs:complexType>
  <xs:choice>
    <xs:element name="ConnectionManager">
      <xs:complexType>
        <xs:sequence/>
        <xs:attribute name="ConnectionString" type="xs:string"
                      use="required" form="qualified"/>
        <xs:attribute name="FileUsageType" type="DTS:FileUsageTypeEnum"
                      default="0" use="optional" form="qualified"/>
      </xs:complexType>
    </xs:element>
  </xs:choice>
</xs:complexType>
```

### 2.4.4.1.3.6   Cache Data Source

When the connection manager is to a Cache data source, the formal type of the **ObjectData** element is the ConnectionManagerObjectDataType type. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type from the restrictions that are declared in the formal type declaration. A **ConnectionManager** instance for a Cache source MUST follow the restrictions that are shown in this anonymous XSD fragment.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **ConnectionManagerObjectDataType** type:

- The value for the **CreationName** attribute of the containing **ConnectionManager** element MUST be "CACHE".

- The **ObjectData** element type MUST be of the type that is specified by the anonymous XSD definition that is contained in this section.

- The **ObjectData** element MUST contain a **ConnectionManager** element.

- The **ConnectionManager** child element to the **ObjectData** element MUST NOT include attributes, except for those that are specified as valid in this anonymous XSD fragment, which is a subset of the attributes that are contained in the ConnectionManagerConnectionManagerAttributeGroup attribute group. The **PropertyExpression** element of the containing **ConnectionManager** element MUST NOT include properties, except for those that are named the same as the attributes for the **ConnectionManager** element in this anonymous type definition XSD fragment.

This anonymous type declaration acts as a replacement for the definition for **ConnectionManagerObjectDataType** type in the case of a Cache data source.

```
<xs:complexType>
  <xs:choice>
    <xs:element name="ConnectionManager">
      <xs:complexType>
        <xs:sequence/>
        <xs:attribute name="UseFile" type="DTS:BooleanStringCap"
                      default="False" use="optional" form="qualified"/>
        <xs:attribute name="UseEncryption" type="DTS:BooleanStringCap"
                      default="False" use="optional" form="qualified"/>
        <xs:attribute name="RetainData" type="DTS:BooleanStringCap"
                      default="True" use="optional" form="qualified"/>
      </xs:complexType>
    </xs:element>
  </xs:choice>
```

```
</xs:complexType>
```

### 2.4.4.1.3.7    Excel Data Source

When the connection manager is to an Excel data source, the formal type of the **ObjectData** element is the ConnectionManagerObjectDataType type. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type from the restrictions that are declared in the formal type declaration. A **ConnectionManager** instance for an Excel source MUST follow the restrictions that are shown in this anonymous XSD fragment.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **ConnectionManagerObjectDataType** type:

▪   The value for the **CreationName** attribute of the containing **ConnectionManager** element MUST be "EXCEL".

▪   The **ObjectData** element type MUST be of the type that is specified by the anonymous XSD definition that is contained in this section.

▪   The **ObjectData** element MUST contain a **ConnectionManager** element.

▪   The **ConnectionManager** child element to the **ObjectData** element MUST NOT include attributes, except for those that are specified as valid in this anonymous XSD fragment, which is a subset of the attributes that are contained in the ConnectionManagerConnectionManagerAttributeGroup attribute group. The **PropertyExpression** element of the containing **ConnectionManager** element MUST NOT include properties, except for those that are named the same as the attributes for the **ConnectionManager** element in this anonymous type definition XSD fragment.

This anonymous type declaration acts as a replacement for the definition for **ConnectionManagerObjectDataType** type in the case of an Excel data source.

```
<xs:complexType>
  <xs:choice>
    <xs:element name="ConnectionManager">
      <xs:complexType>
        <xs:sequence/>
        <xs:attribute name="Retain" type="DTS:BooleanStringCap"
                      default="False" use="optional" form="qualified"/>
        <xs:attribute name="ConnectionString" type="xs:string"
                      use="required" form="qualified"/>
      </xs:complexType>
    </xs:element>
  </xs:choice>
</xs:complexType>
```

### 2.4.4.1.3.8    FTP Data Source

When the connection manager is to an FTP data source, the formal type of the **ObjectData** element is the ConnectionManagerObjectDataType type. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type from the restrictions that are declared in the formal type declaration. A **ConnectionManager** instance for an FTP source MUST follow the restrictions that are shown in this anonymous XSD fragment.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **ConnectionManagerObjectDataType** type:

▪   The value for the **CreationName** attribute of the containing **ConnectionManager** element MUST be "FTP".

- The **ObjectData** element type MUST be of the type that is specified by the anonymous XSD definition that is contained in this section.

- The **ObjectData** element MUST contain a **ConnectionManager** element.

- The **ConnectionManager** child element to the **ObjectData** element MUST NOT include attributes, except for those that are specified as valid in this anonymous XSD fragment, which is a subset of the attributes that are contained in the ConnectionManagerConnectionManagerAttributeGroup attribute group. The **PropertyExpression** element of the containing **ConnectionManager** element MUST NOT include properties, except for those that are named the same as the attributes for the **ConnectionManager** element in this anonymous type definition XSD fragment.

This anonymous type declaration acts as a replacement for the definition for **ConnectionManagerObjectDataType** type in the case of an FTP data source.

```
<xs:complexType>
  <xs:choice>
    <xs:element name="ConnectionManager">
      <xs:complexType>
        <xs:sequence/>
        <xs:attribute name="ConnectionString" type="xs:string"
                      use="required" form="qualified"/>
      </xs:complexType>
    </xs:element>
  </xs:choice>
</xs:complexType>
```

### 2.4.4.1.3.9   HTTP Data Source

When the connection manager is to an HTTP data source, the formal type of the **ObjectData** element is the ConnectionManagerObjectDataType type. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type from the restrictions that are declared in the formal type declaration. A **ConnectionManager** instance for an HTTP data source MUST follow the restrictions that are shown in this anonymous XSD fragment.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **ConnectionManagerObjectDataType** type:

- The value for the **CreationName** attribute of the containing **ConnectionManager** element MUST be "HTTP".

- The **ObjectData** element type MUST be of the type that is specified by the anonymous XSD definition that is contained in this section.

- The **ObjectData** element MUST contain a **ConnectionManager** element.

- The **ConnectionManager** child element to the **ObjectData** element MUST NOT include attributes, except for those that are specified as valid in this anonymous XSD fragment, which is a subset of the attributes that are contained in the ConnectionManagerConnectionManagerAttributeGroup attribute group. The **PropertyExpression** element of the containing **ConnectionManager** element MUST NOT include properties, except for those that are named the same as the attributes for the **ConnectionManager** element in this anonymous type definition XSD fragment.

This anonymous type declaration acts as a replacement for the definition for **ConnectionManagerObjectDataType** type in the case of an HTTP data source.

```
<xs:complexType>
  <xs:choice>
```

```
          <xs:element name="ConnectionManager">
            <xs:complexType>
              <xs:sequence/>
              <xs:attribute name="ConnectionString" type="xs:string"
                            use="required" form="qualified"/>
            </xs:complexType>
          </xs:element>
        </xs:choice>
      </xs:complexType>
```

### 2.4.4.1.3.10  MSMQ Data Source

When the connection manager is to an MSMQ data source, the formal type of the **ObjectData** element is the ConnectionManagerObjectDataType type. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type than the restrictions that are declared in the formal type declaration. A **ConnectionManager** instance to an MSMQ data source MUST follow the restrictions that are shown in this anonymous XSD fragment.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **ConnectionManagerObjectDataType** type:

- The value for the **CreationNameProperty** of the containing **ConnectionManager** element MUST be "MSMQ".

- The **ObjectData** element type MUST be of the type that is specified by the anonymous XSD definition that is contained in this section.

- The **ObjectData** element MUST contain an **MsmqConnectionManager** element of type ConnectionManagerObjectDataMsmqConnectionManagerType.

This anonymous type declaration acts as a replacement for the definition for **ConnectionManagerObjectDataType** type in the case of an MSMQ data source.

```
      <xs:complexType>
        <xs:choice>
          <xs:element name="MsmqConnectionManager"
            type="DTS:ConnectionManagerObjectDataMsmqConnectionManagerType"/>
        </xs:choice>
      </xs:complexType>
```

### 2.4.4.1.3.11  MULTIFILE Data Source

When the connection manager is to a multiple file data source, the formal type of the **ObjectData** element is the ConnectionManagerObjectDataType type. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type from the restrictions that are declared in the formal type declaration. A **ConnectionManager** instance for a multiple file source MUST follow the restrictions that are shown in this anonymous XSD fragment.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **ConnectionManagerObjectDataType** type:

- The value for the **CreationName** attribute of the containing **ConnectionManager** element MUST be "MULTIFILE".

- The **ObjectData** element type MUST be of the type that is specified by the anonymous XSD definition that is contained in this section.

- The **ObjectData** element MUST contain a **ConnectionManager** element.

- The **ConnectionManager** child element to the **ObjectData** element MUST NOT include attributes, except for those that are specified as valid in this anonymous XSD fragment, which is a subset of the attributes that are contained in the ConnectionManagerConnectionManagerAttributeGroup attribute group. The **PropertyExpression** element of the containing **ConnectionManager** element MUST NOT include properties, except for those that are named the same as the attributes for the **ConnectionManager** element in this anonymous type definition XSD fragment.

This anonymous type declaration acts as a replacement for the definition for **ConnectionManagerObjectDataType** type in the case of a multiple file data source.

```
<xs:complexType>
  <xs:choice>
    <xs:element name="ConnectionManager">
      <xs:complexType>
        <xs:sequence/>
        <xs:attribute name="ConnectionString" type="xs:string"
                      use="required" form="qualified"/>
        <xs:attribute name="FileUsageType" type="DTS:FileUsageTypeEnum"
                      default="0" use="optional" form="qualified"/>
      </xs:complexType>
    </xs:element>
  </xs:choice>
</xs:complexType>
```

### 2.4.4.1.3.12 MULTIFLATFILE Data Source

When the connection manager is to a multiple flat file data source, the formal type of the **ObjectData** element is the ConnectionManagerObjectDataType type. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type from the restrictions that are declared in the formal type declaration. A **ConnectionManager** instance for a multiple flat file source MUST follow the restrictions that are shown in this anonymous XSD fragment.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **ConnectionManagerObjectDataType** type:

- The value for the **CreationName** attribute of the containing **ConnectionManager** element MUST be "MULTIFLATFILE".

- The **ObjectData** element type MUST be of the type that is specified by the anonymous XSD definition that is contained in this section.

- The **ObjectData** element MUST contain a **ConnectionManager** element.

- The **ConnectionManager** child element to the **ObjectData** element MUST NOT include attributes, except for those that are specified as valid in this anonymous XSD fragment, which is a subset of the attributes that are contained in the ConnectionManagerConnectionManagerAttributeGroup attribute group. The **PropertyExpression** element of the containing **ConnectionManager** element MUST NOT include properties, except for those that are named the same as the attributes for the **ConnectionManager** element in this anonymous type definition XSD fragment.

This anonymous type declaration acts as a replacement for the definition for **ConnectionManagerObjectDataType** type in the case of a multiple flat file data source.

```
<xs:complexType>
  <xs:choice>
    <xs:element name="ConnectionManager">
      <xs:complexType>
        <xs:sequence/>
        <xs:attribute name="ConnectionString" type="xs:string"
```

```
                  use="required" form="qualified"/>
          <xs:attribute name="FileUsageType" type="DTS:FileUsageTypeEnum"
                  default="0" use="optional" form="qualified"/>
      </xs:complexType>
    </xs:element>
  </xs:choice>
</xs:complexType>
```

### 2.4.4.1.3.13 ODBC Data Source

When the connection manager is to an ODBC data source, the formal type of the **ObjectData** element is the ConnectionManagerObjectDataType type. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type from the restrictions that are declared in the formal type declaration. A **ConnectionManager** instance for an ODBC source MUST follow the restrictions that are shown in this anonymous XSD fragment.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **ConnectionManagerObjectDataType** type:

- The value for the **CreationName** attribute of the containing **ConnectionManager** element MUST be "ODBC".

- The **ObjectData** element type MUST be of the type that is specified by the anonymous XSD definition that is contained in this section.

- The **ObjectData** element MUST contain a **ConnectionManager** element.

- The **ConnectionManager** child element to the **ObjectData** element MUST NOT include attributes, except for those that are specified as valid in this anonymous XSD fragment, which is a subset of the attributes that are contained in the ConnectionManagerConnectionManagerAttributeGroup attribute group. The **PropertyExpression** element of the containing **ConnectionManager** element MUST NOT include properties, except for those that are named the same as the attributes for the **ConnectionManager** element in this anonymous type definition XSD fragment.

This anonymous type declaration acts as a replacement for the definition for **ConnectionManagerObjectDataType** type in the case of an ODBC data source.

```
<xs:complexType>
  <xs:choice>
    <xs:element name="ConnectionManager">
      <xs:complexType>
        <xs:sequence/>
        <xs:attribute name="Retain" type="DTS:BooleanStringCap"
                  default="False" use="optional" form="qualified"/>
        <xs:attribute name="ConnectionString" type="xs:string"
                  use="required" form="qualified"/>
        <xs:attribute name="ServerName" type="xs:string" use="required"
                  form="qualified"/>
      </xs:complexType>
    </xs:element>
  </xs:choice>
</xs:complexType>
```

### 2.4.4.1.3.14 SMOServer Data Source

When the connection manager is to an SMOServer data source, the formal type of the **ObjectData** element is the ConnectionManagerObjectDataType type. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type than the restrictions that are declared in the formal type declaration. A **ConnectionManager** instance to an SMOServer source MUST follow the restrictions that are shown in this anonymous XSD fragment.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **ConnectionManagerObjectDataType** type:

- The value for the **CreationName Property** of the containing **ConnectionManager** element MUST be "SMOServer".

- The **ObjectData** element type MUST be of the type that is specified by the anonymous XSD definition that is contained in this section.

- The **ObjectData** element MUST contain an **SMOServerConnectionManager** element of type ConnectionManagerObjectDataSMOServerConnectionManagerType.

This anonymous type declaration acts as a replacement for the definition for **ConnectionManagerObjectDataType** type in the case of an SMOServer data source.

```
<xs:complexType>
  <xs:choice>
    <xs:element name="SMOServerConnectionManager"
      type="DTS:ConnectionManagerObjectDataSMOServerConnectionManagerType"/>
  </xs:choice>
</xs:complexType>
```

### 2.4.4.1.3.15  SMTP Data Source

When the connection manager is to an SMTP data source, the formal type of the **ObjectData** element is the ConnectionManagerObjectDataType type. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type than the restrictions that are declared in the formal type declaration. A **ConnectionManager** instance to an SMTP data source MUST follow the restrictions that are shown in this anonymous XSD fragment.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **ConnectionManagerObjectDataType** type:

- The value for the **CreationName Property** of the containing **ConnectionManager** element MUST be "SMOServer".

- The **ObjectData** element type MUST be of the type that is specified by the anonymous XSD definition that is contained in this section.

- The **ObjectData** element MUST contain an **SmtpConnectionManager** element of type ConnectionManagerObjectDataSmtpConnectionManagerType.

This anonymous type declaration acts as a replacement for the definition for **ConnectionManagerObjectDataType** type in the case of an SMTP data source.

```
<xs:complexType>
  <xs:choice>
    <xs:element name="SmtpConnectionManager"
      type="DTS:ConnectionManagerObjectDataSmtpConnectionManagerType"/>
  </xs:choice>
</xs:complexType>
```

### 2.4.4.1.3.16  SQLMOBILE Data Source

When the connection manager is to a SQL Server Mobile data source, the formal type of the **ObjectData** element is the ConnectionManagerObjectDataType type. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type from the restrictions that are declared in the formal type declaration. A

**ConnectionManager** instance for a SQL Server Mobile source MUST follow the restrictions that are shown in this anonymous XSD fragment.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **ConnectionManagerObjectDataType** type:

- The value for the **CreationName** attribute of the containing **ConnectionManager** element MUST be "SQLMOBILE".

- The **ObjectData** element type MUST be of the type that is specified by the anonymous XSD definition that is contained in this section.

- The **ObjectData** element MUST contain a **ConnectionManager** element.

- The **ConnectionManager** child element to the **ObjectData** element MUST NOT include attributes, except for those that are specified as valid in this anonymous XSD fragment, which is a subset of the attributes that are contained in the ConnectionManagerConnectionManagerAttributeGroup attribute group. The **PropertyExpression** element of the containing **ConnectionManager** element MUST NOT include properties, except for those that are named the same as the attributes for the **ConnectionManager** element in this anonymous type definition XSD fragment.

This anonymous type declaration acts as a replacement for the definition for **ConnectionManagerObjectDataType** type in the case of a SQL Server Mobile data source.

```
<xs:complexType>
  <xs:choice>
    <xs:element name="ConnectionManager">
      <xs:complexType>
        <xs:sequence/>
        <xs:attribute name="ConnectionString" type="xs:string"
                      use="required" form="qualified"/>
      </xs:complexType>
    </xs:element>
  </xs:choice>
</xs:complexType>
```

### 2.4.4.1.3.17  WMI Data Source

When the connection manager is to a WMI data source, the formal type of the **ObjectData** element is the ConnectionManagerObjectDataType type. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type than the restrictions that are declared in the formal type declaration. A **ConnectionManager** instance to a WMI data source MUST follow the restrictions that are shown in this anonymous XSD fragment.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **ConnectionManagerObjectDataType** type:

- The value for the **CreationName Property** of the containing **ConnectionManager** element MUST be "WMI".

- The **ObjectData** element type MUST be of the type that is specified by the anonymous XSD definition that is contained in this section.

- The **ObjectData** element MUST contain a **WmiConnectionManager** element of type ConnectionManagerObjectDataWmiConnectionManagerType.

This anonymous type declaration acts as a replacement for the definition for **ConnectionManagerObjectDataType** type in the case of a WMI data source.

```
<xs:complexType>
  <xs:choice>
    <xs:element name="WmiConnectionManager"
      type="DTS:ConnectionManagerObjectDataWmiConnectionManagerType"/>
  </xs:choice>
</xs:complexType>
```

## 2.4.5 ExecutableTypePackagePropertyNameEnum

The **ExecutableTypePackagePropertyNameEnum** type contains the values for the **Name** attribute of the property elements that are allowed for executable type ExecutableTypePackage.

The following is the XSD for the **ExecutableTypePackagePropertyNameEnum** type.

```
<xs:simpleType name="ExecutableTypePackagePropertyNameEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="PackageFormatVersion"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **ExecutableTypePackagePropertyNameEnum** type.

| Enumeration value | Description |
| --- | --- |
| PackageFormatVersion | Specifies the package format version. |

## 2.4.6 ExecutableTypePackageAttributeGroup

The **ExecutableTypePackageAttributeGroup** attribute group contains the attributes for the ExecutableTypePackage type.

The following is the XSD for the **ExecutableTypePackageAttributeGroup** attribute group.

```
<xs:attributeGroup name="ExecutableTypePackageAttributeGroup">
  <xs:attribute name="VersionComments" type="xs:string" default=""
                use="optional" form="qualified"/>
  <xs:attribute name="CreatorName" type="xs:string" use="optional"
                form="qualified"/>
  <xs:attribute name="CreatorComputerName" type="xs:string"
                use="optional" form="qualified"/>
  <xs:attribute name="CreationDate" type="xs:dateTime" use="optional"
                form="qualified"/>
  <xs:attribute name="PackageType" type="DTS:PackageTypeEnum" default="0"
                use="optional" form="qualified"/>
  <xs:attribute name="ProtectionLevel" type="DTS:ProtectionLevelEnum"
                default="1" use="optional" form="qualified"/>
  <xs:attribute name="MaxConcurrentExecutables" type="xs:int"
                default="-1" use="optional" form="qualified"/>
  <xs:attribute name="PackagePriorityClass" default="0" use="optional"
                type="DTS:PackagePriorityClassEnum" form="qualified"/>
  <xs:attribute name="VersionMajor" type="xs:int" default="1"
                use="optional" form="qualified"/>
  <xs:attribute name="VersionMinor" type="xs:int" default="0"
                use="optional" form="qualified"/>
  <xs:attribute name="VersionBuild" type="xs:int" default="0"
                use="optional" form="qualified"/>
  <xs:attribute name="VersionGUID" type="DTS:uuid" use="optional"
```

```
                            form="qualified"/>
    <xs:attribute name="EnableConfig" type="DTS:BooleanStringCap"
                            default="False" use="optional" form="qualified"/>
    <xs:attribute name="CheckpointFileName" type="xs:string" default=""
                            use="optional" form="qualified"/>
    <xs:attribute name="SaveCheckpoints" type="DTS:BooleanStringCap"
                            default="False" use="optional" form="qualified"/>
    <xs:attribute name="CheckpointUsage" type="DTS:CheckpointUsageEnum"
                            default="0" use="optional" form="qualified"/>
    <xs:attribute name="SuppressConfigurationWarnings" default="False"
                            type="DTS:BooleanStringCap" use="optional"
                            form="qualified"/>
    <xs:attribute name="LastModifiedProductVersion" type="xs:string"
                            use="optional" form="qualified"/>
</xs:attributeGroup>
```

The following table specifies the attributes for the **ExecutableTypePackageAttributeGroup** attribute group.

| Attribute | Description |
|---|---|
| VersionComments | A string value that specifies text that can be used to store comments about the version.<12> |
| CreatorName | A string value that specifies text that can be used to store the creator name of the package as documentation.<13> |
| CreatorComputerName | A string value that specifies text that serves as documentation of the computer on which the package was created.<14> |
| CreationDate | A date/time value that specifies the date that the executable was created.<15> |
| PackageType | An enumeration value that specifies how the package was created.<16> |
| ProtectionLevel | An enumeration value that specifies how sensitive information is saved with packages. |
| MaxConcurrentExecutables | An integer value that specifies the maximum number of executables to execute concurrently in the package. Valid values are -1 and any integer greater than or equal to 1. <br> A value of -1 allows the maximum number of concurrently running executables to equal the number of processors plus two. |
| PackagePriorityClass | An enumeration value that specifies the execution priority of the package. |
| VersionMajor | An integer value that specifies the major version of the package. Valid values are integers greater than or equal to 0. |
| VersionMinor | An integer value that specifies the minor version of the package. Valid values are integers greater than or equal to 0. |
| VersionBuild | An integer value that specifies the build number of the package. Valid values are integers greater than or equal to 0.<17> |
| VersionGUID | A value of type **DTS:uuid** that specifies the globally unique identifier (GUID) of the package version.<18> |
| EnableConfig | A value of type **DTS:BooleanStringCap** that specifies whether settings from external configuration settings are accessible to the executable. <br> "True" specifies that configurations are used with the package. <br> "False" specifies that configurations are not used with the package. |

| Attribute | Description |
|---|---|
| CheckpointFileName | A string value that specifies the name of a checkpoint file that a checkpoint-enabled package uses. |
| SaveCheckpoints | A value of type **DTS:BooleanStringCap** that specifies whether the package saves checkpoints.<br><br>"True" specifies that saving of checkpoints is enabled.<br><br>"False" specifies that saving of checkpoints is not enabled. |
| CheckpointUsage | An enumeration value that specifies when the package uses checkpoints. |
| SuppressConfigurationWarnings | A value of type **DTS:BooleanStringCap** that specifies whether configuration warnings are suppressed.<br><br>"True" specifies that configuration warnings are suppressed.<br><br>"False" specifies that configuration warnings are not suppressed. |
| LastModifiedProductVersion | A string value that specifies the latest version of the product that is installed.<19> |

## 2.4.6.1 PackageTypeEnum

The **PackageTypeEnum** type contains the enumeration values that specify how the package was created.

The following is the XSD of the **PackageTypeEnum** type.

```
<xs:simpleType name="PackageTypeEnum">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="6"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the descriptions for the enumeration values of the **PackageTypeEnum** type.

| Enumeration value | Description |
|---|---|
| 0 | Default |
| 1 | DTS Wizard |
| 2 | DTS Designer |
| 3 | SQL Replication |
| 5 | DTS Designer 100 |
| 6 | SQL DB Maintenance |

### 2.4.6.2 ProtectionLevelEnum

The **ProtectionLevelEnum** type contains the enumeration values that specify how sensitive information is saved with the package.

The following is the XSD of the **ProtectionLevelEnum** type.

```
<xs:simpleType name="ProtectionLevelEnum">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="5"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the descriptions for the enumeration values of the **ProtectionLevelEnum** type.

| Enumeration value | Description |
|---|---|
| 0 | Do not save sensitive information. |
| 1 | Encrypt sensitive information with user key. |
| 2 | Encrypt sensitive information with password. |
| 3 | Encrypt all information with password. |
| 4 | Encrypt all information with user key. |
| 5 | Use server storage. |

### 2.4.6.3 PackagePriorityClassEnum

The **PackagePriorityClassEnum** type contains the enumeration values that specify how sensitive information is saved with the package.

The following is the XSD of the **PackagePriorityClassEnum** type.

```
<xs:simpleType name="PackagePriorityClassEnum">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="4"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the descriptions for the enumeration values of the **PackagePriorityClassEnum** type.

| Enumeration value | Priority for package |
|---|---|
| 0 | Default |
| 1 | Above normal |
| 2 | Normal |
| 3 | Below normal |

| Enumeration value | Priority for package |
|---|---|
| 4 | Idle |

### 2.4.6.4 CheckpointUsageEnum

The **CheckpointUsageEnum** type contains the enumeration values that specify when the package uses checkpoints.

The following is the XSD of the **CheckpointUsageEnum** type.

```
<xs:simpleType name="CheckpointUsageEnum">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="2"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the descriptions for the enumeration values of the **CheckpointUsageEnum** type.

| Enumeration value | Description |
|---|---|
| 0 | Never use checkpoints. |
| 1 | Use checkpoints if they exist. |
| 2 | Always use checkpoints. |

### 2.4.7 PackageParametersType

The **PackageParametersType** complex type is the container type for a collection of elements of type PackageParameterType.

The following is the XSD of the **PackageParametersType** type.

```
<xs:complexType name="PackageParametersType">
  <xs:sequence>
    <xs:element name="PackageParameter" type="DTS:PackageParameterType" minOccurs="0"
 maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
```

The following table provides additional information about the elements, types, and constraints for the **PackageParametersType** complex type.

| Element | Type definition | Description |
|---|---|---|
| PackageParameter | PackageParameterType | Specifies a **PackageParameter** element that contains the values and content for a package parameter. |

### 2.4.7.1 PackageParameterType

The **PackageParameterType** complex type is used to specify package parameters.

The following is the XSD for the **PackageParameterType** complex type.

```
    <xs:complexType name="PackageParameterType">
      <xs:sequence>
        <xs:element maxOccurs="unbounded" name="Property"
 type="DTS:PackageParameterPropertyType" />
      </xs:sequence>
      <xs:attribute name="CreationName" use="required" type="xs:string" />
      <xs:attribute name="DataType" type="xs:unsignedByte" use="required" />
      <xs:attribute name="Description" type="xs:string" use="optional" />
      <xs:attribute name="DTSID" type="DTS:uuid" use="required" />
      <xs:attribute name="ObjectName" type="xs:string" use="required" />
      <xs:attribute name="Required" type="xs:string" use="optional" />
      <xs:attribute name="Sensitive" type="xs:string" use="optional" />
    </xs:complexType>
```

The following table provides additional information about the attributes of the **PackageParameterType** type.

| Attribute | Constraints | Description |
|---|---|---|
| CreationName | SHOULD be empty. | This attribute is ignored. |
| DataType | MUST be an integer value from the following table. | Specifies the data type of the parameter. |
| Description | None. | A description of the parameter. |
| DTSID | None. | A universally unique identifier (UUID) that uniquely identifies the package parameter. |
| ObjectName | MUST be a valid name. | The name of the parameter. |
| Required | MUST be a value that is either the string literal True or the string literal False. | If present and having the value True, the parameter is declared to be a required parameter. |
| Sensitive | MUST be a value that is either the string literal True or the string literal False. | If present and having the value True, the parameter is declared to be a sensitive parameter |

The following table lists the allowable values for the **DataType** attribute and the data types that correspond to those values.

| Integer value | Corresponding data type in the Microsoft .NET Framework |
|---|---|
| 2 | Int16 |
| 3 | Int32 |
| 4 | Single |
| 5 | Double |
| 7 | DateTime |
| 8 | String |
| 11 | Boolean |

| Integer value | Corresponding data type in the Microsoft .NET Framework |
|---|---|
| 14 | Decimal |
| 16 | SByte |
| 17 | Byte |
| 19 | UInt32 |
| 20 | Int64 |
| 21 | UInt64 |

### 2.4.7.2 PackageParameterPropertyType

The **PackageParameterPropertyType** complex type is used to specify the properties of package parameters. In particular, the value of a non-sensitive parameter is stored in a property child element of the **PackageParameter** element representing that parameter.

The following is the XSD for the **PackageParameterPropertyType** complex type

```
    <xs:complexType mixed="true" name="PackageParameterPropertyType">
      <xs:sequence minOccurs="0">
        <xs:element name="Property"
 type="DTS:PackageParameterPropertyEncryptedValuePropertyType" />
      </xs:sequence>
      <xs:attribute name="DataType" type="xs:unsignedByte" use="optional" />
      <xs:attribute name="Name" type="xs:string" use="required" />
    </xs:complexType>
```

The following table provides additional information about the attributes of the **PackageParameterPropertyType**.

| Attribute | Constraints | Description |
|---|---|---|
| DataType | If present, MUST be the same as the value of the **DataType** attribute of the PackageParameterType element that contains the **PackageParameterPropertyType** element. MUST be present when the **Sensitive** attribute of the **PackageParameterType** element that contains the **PackageParameterPropertyType** element is not present or has the value False. MUST NOT be present when the value of that attribute is True. | Specifies the data type of the property. |
| Name | MUST be the value DefaultValue. | Specifies the name of the property. |

If the **Sensitive** attribute of the **PackageParameterType** element that contains the **PackageParameterPropertyType** element has the value True, this element MUST contain a child element of type PackageParameterEncryptedValuePropertyType and MUST NOT contain a simple value.

### 2.4.7.3 PackageParameterEncryptedValuePropertyType

The **PackageParameterEncryptedValuePropertyType** complex type is used to specify the value of a sensitive package parameter.

The following is the XSD for the **PackageParameterEncryptedValuePropertyType** complex type.

```
<xs:complexType name="PackageParameterPropertyEncryptedValuePropertyType">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="DataType" type="xs:unsignedByte" use="required" />
      <xs:attribute name="Name" type="xs:string" use="required" />
      <xs:attribute name="Encrypted" type="xs:unsignedByte" use="required" />
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

The following table provides additional information about the attributes of the **PackageParameterEncryptedValuePropertyType**.

| Attribute | Constraints | Description |
|---|---|---|
| DataType | MUST be the same as the value of the **DataType** attribute of the PackageParameterType element that contains the **PackageParameterEncryptedValuePropertyType** element. MUST be present when the **Sensitive** attribute of the **PackageParameterType** element that contains the **PackageParameterEncryptedValuePropertyType** element is not present or has the value False. MUST NOT be present when the value of that attribute is True. | Specifies the data type of the property. |
| Name | MUST be the value DefaultValue. | Specifies the name of the property. |
| Encrypted | MUST be the value 1. | Specifies that the value is encrypted. |

## 2.5   ExecutablesType

The **ExecutablesType** complex type is the container type for a collection of elements of type AnyNonPackageExecutableType.

The following is the XSD of the **ExecutablesType** complex type.

```
<xs:complexType name="ExecutablesType">
  <xs:sequence>
    <xs:element name="Executable" type="DTS:AnyNonPackageExecutableType" minOccurs="0"
 maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

The following table provides additional information about the elements, types, and constraints for the **ExecutablesType** complex type.

| Element | Type definition | Description |
|---|---|---|
| Executable | AnyNonPackageExecutableType | Specifies an **Executable** element that in turn specifies an executable to execute along with the options, parameters, and settings for the executable. |

## 2.5.1 AnyNonPackageExecutableType

The **AnyNonPackageExecutableType** complex type is the type of an Executable element that is contained within another **Executable** element. That is, it is the type definition for any **Executable** element except the **Executable** element that is the root node of the package.

The following is the XSD for the **AnyNonPackageExecutableType** complex type.

```
<xs:complexType name="AnyNonPackageExecutableType">
  <xs:sequence>
    <xs:element name="ForEachEnumerator"
                type="DTS:ForEachEnumeratorType" minOccurs="0"/>
    <xs:element name="Variables" type="DTS:VariablesType" minOccurs="0"/>
    <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType" minOccurs="0"/>
    <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
                minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Executables" type="DTS:ExecutablesType" minOccurs="0"/>
    <xs:element name="PrecedenceConstraints" type="DTS:PrecedenceConstraintsType"
                minOccurs="0"/>
    <xs:element name="ForEachVariableMappings"
                type="DTS:ForEachVariableMappingsType" minOccurs="0"/>
    <xs:element name="EventHandlers" type="DTS:EventHandlersType" minOccurs="0"/>
    <xs:element name="ObjectData" type="DTS:ExecutableObjectDataType" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="ExecutableType" use="required" type="xs:string"/>
  <xs:attribute name="ThreadHint" use="optional" type="xs:int"/>
  <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
  <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
  <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
  <xs:attributeGroup ref="DTS:AnyNonPackageExecutableAttributeGroup"/>
  <xs:attribute name="refId" type="xs:string" use="required" form="qualified"/>
</xs:complexType>
```

The following table provides additional information about the elements, types, and constraints in the **AnyNonPackageExecutableType** complex type.

| Element | Constraints | Type definition | Description |
|---|---|---|---|
| ForEachEnumerator | MUST NOT be used unless the **ExecutableType** attribute is equal to "STOCK:FOREACHLOOP" | ForEachEnumeratorType | Specifies a collection of elements of type **ForEachEnumeratorType**. Each such element defines the iterations of a **For...Each** loop that is available to the executable. |
| Variables | None | VariablesType | Specifies a collection of elements of type **VariableType**. Each such element defines a variable that is available to the executable. |
| LoggingOptions | None | LoggingOptionsType | Specifies information about what kind of information is logged for this package and where it is recorded. |
| PropertyExpression | Additional constraints exist for the **PropertyExpression** element. | PropertyExpressionElementType | Serves the same purpose as specified for the **Property** element except that an expression that is evaluated at run time is contained in the element's value. |

| Element | Constraints | Type definition | Description |
|---|---|---|---|
| Executables | None | ExecutablesType | Specifies a collection of elements of type **AnyNonPackageExecutableType**. Each such element defines an executable that is available to the executable. |
| PrecedenceConstraints | None | PrecedenceConstraintsType | Specifies a collection of elements of type **PrecedenceConstraintType**. Each such element defines a precedence constraint that is available to the executable. |
| ForEachVariableMappings | MUST NOT be used unless the **ExecutableType** attribute is equal to "STOCK:FOREACHLOOP" | ForEachVariableMappingsType | Specifies a collection of elements of type **ForEachVariableMappingType**. Each such element defines variable mappings for the iterations of a **For...Each** loop that are available to the executable that contains the collection. |
| EventHandlers | None | EventHandlersType | Specifies a collection of elements of type **EventHandlerType**. Each such element defines an event handler that is available to the executable. |
| ObjectData | Each value of the attribute **ExecutableType** constrains the choice that is contained in the definition of ExecutableObjectDataType to a specific choice. For more information, see ExecutableType Attribute Values for Different Executables. | ExecutableObjectDataType | Contains information specific to the **Executable**. The nature of the information varies with the value of the **ExecutableType** attribute. For more information, see Executable ObjectData Types. |

The following table provides additional information about the attributes, attribute groups, and types in the **AnyNonPackageExecutableType** complex type.

| Attribute | Type definition | Description |
|---|---|---|
| ExecutableType | Some values of the **ExecutableType** attribute allow further **Executable** elements to be contained, and other values MUST NOT contain further **Executable** elements. For more information, see ExecutableType Attribute Values for Different Executables. | ExecutableTypePackage |
| ThreadHint | Not all values of **ExecutableType** are allowed to have the **ThreadHint** | An integer value.[20] |

| Attribute | Type definition | Description |
|---|---|---|
| | | attribute present. For more information, see ExecutableType Attribute Values for Different Executables. |
| refId | | A string value that specifies a unique **refId** for this object. Elsewhere within the containing executable or within the package, the value of **refId** can be used to refer to this object.[21] |
| AllExecutableAttributeGroup | AllExecutableAttributeGroup | An attribute group that specifies attributes that are common to many different types throughout the schema. |
| BaseExecutablePropertyAttributeGroup | BaseExecutablePropertyAttributeGroup | An attribute group that specifies attributes that are common to many different types throughout the schema. |
| BasePropertyAttributeGroup | BasePropertyAttributeGroup | An attribute group that specifies attributes that are common to many different types throughout the schema. |
| AnyNonPackageExecutableAttributeGroup | AnyNonPackageExecutableAttributeGroup | An attribute group that specifies attributes for the **AnyNonPackageExecutableType** type. |

### 2.5.1.1 ForEachEnumeratorType

The **ForEachEnumeratorType** complex type MUST NOT be used for an executable unless its **ExecutableType** attribute is equal to "STOCK:FOREACHLOOP". The **ForEachEnumeratorType** complex type contains information about the entity to be looped over and the logic of the looping.

The following is the XSD for the **ForEachEnumeratorType** complex type.

```
<xs:complexType name="ForEachEnumeratorType">
  <xs:sequence>
    <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
                minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="ObjectData" type="DTS:ForEachEnumeratorObjectDataType"/>
  </xs:sequence>
  <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
</xs:complexType>
```

The following table provides additional information about the elements, types, and constraints in the **ForEachEnumeratorType** complex type.

| Element | Additional constraints | Type definition | Description |
|---|---|---|---|
| PropertyExpression | Additional constraints | PropertyExpressionElementType | Serves the same purpose |

| Element | Additional constraints | Type definition | Description |
|---|---|---|---|
| | exist for the **PropertyExpression** element. These constraints MUST be followed. | | as described for the **Property** element, except that an expression that is evaluated at run time is contained in the element's value that is stored in the Integration Services file format. |
| ObjectData | None | ForEachEnumeratorObjectDataType | Contains information that is specific to the **ForEachEnumerator**. For more information, see Executable ObjectData Types. |

The following table provides additional information about the attributes, attribute groups, and types for the **ForEachEnumeratorType** complex type.

| Attribute | Description |
|---|---|
| BasePropertyAttributeGroup | An attribute group that specifies attributes that are common to many different types throughout the schema. |

The **ForEachEnumeratorType** complex type has no additional attributes or attribute groups beyond those of the **BasePropertyAttributeGroup** attribute group. Therefore, no further attribute groups are described in any of this section's subsections.

### 2.5.1.1.1 ForEachEnumeratorObjectDataType

The **ForEachEnumeratorObjectDataType** complex type contains information that is specific to each type of **For...Each** loop.

The following is the XSD of the **ForEachEnumeratorObjectDataType** complex type.

```
<xs:complexType name="ForEachEnumeratorObjectDataType">
  <xs:choice>
    <xs:element name="FEIEItems" type="DTS:FEIEItemsType" form="unqualified"/>
    <xs:element name="ForEachFileEnumeratorProperties"
                type="DTS:ForEachFileEnumeratorPropertiesType" form="unqualified"/>
    <xs:element name="FEEADO" type="DTS:FEEADOType" form="unqualified"/>
    <xs:element name="FEESchemaRowset" type="DTS:FEESchemaRowsetType"
                form="unqualified"/>
    <xs:element name="FEEFVE" type="DTS:FEEFVEType" form="unqualified"/>
    <xs:element name="FEENODELIST" type="DTS:FEENODELISTType" form="unqualified"/>
    <xs:element name="FEESMO" type="DTS:FEESMOType" form="unqualified"/>
  </xs:choice>
</xs:complexType>
```

The following table provides additional information about the elements, types, and constraints in the **ForEachEnumeratorObjectDataType** complex type.

| Element | Additional constraints | Type definition | Description |
|---|---|---|---|
| FEIEItems | None | FEIEItemsType | Specifies a collection of items over which enumeration is |

| Element | Additional constraints | Type definition | Description |
|---|---|---|---|
| | | | performed. |
| ForEachFileEnumeratorProperties | None | ForEachFileEnumeratorPropertiesType | Specifies properties for enumeration over a set of files. |
| FEEADO | None | FEEADOType | Specifies properties and parameters for enumeration over the rows of an ADO recordset or over an ADO.NET dataset. |
| FEESchemaRowset | None | FEESchemaRowsetType | Specifies properties and parameters for enumeration over the tables of a database schema. |
| FEEFVE | None | FEEFVEType | Specifies properties and parameters for enumeration over the content of a variable in the package. |
| FEENODELIST | None | FEENODELISTType | Specifies properties and parameters for enumeration over the nodes in an XML document. |
| FEESMO | None | FEESMOType | Specifies properties and parameters for enumeration over tables in a database schema using an SMO Server connection. |

### 2.5.1.1.1.1    FEIEItemsType

The **FEIEItemsType** complex type specifies a collection of items over which enumeration is performed.

The following is the XSD of the **FEIEItemsType** complex type.

```
<xs:complexType name="FEIEItemsType">
  <xs:sequence>
    <xs:element name="FEIEItem" type="DTS:FEIEItemType"
              maxOccurs="unbounded" form="unqualified"/>
  </xs:sequence>
</xs:complexType>
```

The following table provides additional information about the elements, types, and constraints in the **FEIEItemsType** complex type.

| Element | Additional constraints | Type definition | Description |
|---------|------------------------|-----------------|-------------|
| FEIEItem | None | FEIEItemType | Specifies an individual item within the collection of items over which enumeration is being performed. |

### 2.5.1.1.1.1.1 FEIEItemType

The **FEIEItemType** complex type specifies an individual item within a collection of items over which enumeration is performed.

The following is the XSD of the **FEIEItemType** complex type.

```
<xs:complexType name="FEIEItemType">
  <xs:sequence>
    <xs:element name="FEIEItemValue" maxOccurs="unbounded"
                form="unqualified" type="DTS:FEIEItemValueType"/>
  </xs:sequence>
</xs:complexType>
```

The following table provides additional information about the elements, types, and constraints in the **FEIEItemType** complex type.

| Element | Additional constraints | Type definition | Description |
|---------|------------------------|-----------------|-------------|
| FEIEItemValue | None | FEIEItemValueType | Contains the type of and the value for an individual item in the enumeration collection. |

### 2.5.1.1.1.1.1.1 FEIEItemValue Type

The **FEIEItemValueType** complex type specifies the value for an individual FEIEItem item.

The following is the XSD of the **FEIEItemValueType** complex type.

```
<xs:complexType name="FEIEItemValueType">
  <xs:sequence/>
  <xs:attribute name="Type" type="DTS:DtsDataTypeEnum" use="required"
form="unqualified"/>
  <xs:attribute name="Value" type="xs:string" use="required" form="unqualified"/>
</xs:complexType>
```

The following table provides additional information about the attributes of the **FEIEItemValueType** complex type.

| Attribute | Description |
|-----------|-------------|
| Type | The type of the current instance of the **FEIEItemValue** element. |
| Value | The value of the current instance of the **FEIEItemValue** element. The value of the **Value** item is interpreted as being of the type that is specified in the **Type** attribute. |

### 2.5.1.1.1.2 ForEachFileEnumeratorPropertiesType

The **ForEachFileEnumeratorPropertiesType** complex type contains the information for enumeration over a set of files.

The following is the XSD of the **ForEachFileEnumeratorPropertiesType** complex type.

```
<xs:complexType name="ForEachFileEnumeratorPropertiesType">
  <xs:sequence>
    <xs:element name="FEFEProperty" maxOccurs="unbounded" form="unqualified">
      <xs:complexType>
        <xs:sequence/>
        <xs:attribute name="Folder" form="unqualified"/>
        <xs:attribute name="FileSpec" form="unqualified"/>
        <xs:attribute name="FileNameRetrievalType" form="unqualified">
          <xs:simpleType>
            <xs:restriction base="xs:int">
              <xs:minInclusive value="0"/>
              <xs:maxInclusive value="2"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="Recurse" form="unqualified">
          <xs:simpleType>
            <xs:restriction base="xs:int">
              <xs:minInclusive value="-1"/>
              <xs:maxInclusive value="0"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

The following table provides additional information about the attributes for the **FEFEProperty** element of the **ForEachFileEnumeratorPropertiesType** complex type.

| Attribute | Additional constraints | Description |
|---|---|---|
| Folder | None | Specifies a string that contains a folder name over which to enumerate files. |
| FileSpec | None | Specifies a string that contains a file specification. |
| FileNameRetrievalType | None | Specifies the nature of the file name retrieval specification, according to the following integer values: 0 specifies a fully qualified file name. 1 specifies a file name and a file name extension. 2 specifies only a file name. |
| Recurse | None | Specifies whether to traverse all subfolders to search for files that match the file specification according to the following integer values: -1 specifies TRUE; do traverse subfolders. 0 specifies FALSE; do not traverse subfolders. |

### 2.5.1.1.1.3 FEEADOType

The **FEEADOType** complex type specifies enumeration over the rows of an ADO recordset or over an ADO.NET dataset.

The following is the XSD of the **FEEADOType** complex type.

```
<xs:complexType name="FEEADOType">
  <xs:sequence/>
  <xs:attribute name="EnumType" use="required" form="unqualified"
                type="DTS:FEEADOEnumType"/>
  <xs:attribute name="VarName" type="xs:string" use="required" form="unqualified"/>
</xs:complexType>
```

The following table provides additional information about the attributes of the **FEEADOType** complex type.

| Attribute | Additional constraints | Description |
|---|---|---|
| EnumType | None | Specifies the type of enumeration to perform over the tables of a database or over the tables of an ADO.NET dataset. |
| VarName | None | Specifies a string that contains the name of the variable that contains the results over which to perform the enumeration. |

### 2.5.1.1.1.3.1 FEEADOEnumType

The **FEEADOEnumType** type specifies the enumeration values for the **FEEADO** element.

The following is the XSD of the **FEEADOEnumType** type.

```
<xs:simpleType name="FEEADOEnumType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="EnumerateRowsInFirstTable"/>
    <xs:enumeration value="EnumerateAllRows"/>
    <xs:enumeration value="EnumerateTables"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **NodelistEnumerationTypeEnum** type.

| Enumeration value | Description |
|---|---|
| EnumerateRowsInFirstTable | Enumerate over the row values in only the first table. |
| EnumerateAllRows | Enumerate over all rows in all tables. |
| EnumerateTables | Enumerate over each table. |

### 2.5.1.1.1.4 FEESchemaRowsetType

The **FEESchemaRowsetType** complex type specifies enumeration over tables in a database schema.

The following is the XSD of the **FEESchemaRowsetType** complex type.

```xsd
<xs:complexType name="FEESchemaRowsetType">
  <xs:sequence>
    <xs:element name="Restriction" maxOccurs="unbounded"
                form="unqualified" type="DTS:RestrictionElementType"/>
  </xs:sequence>
  <xs:attribute name="Connection" type="xs:string" use="required" form="unqualified"/>
  <xs:attribute name="Schema" type="DTS:uuid" use="required" form="unqualified"/>
</xs:complexType>
```

The following table provides additional information about the elements, types, and constraints in the **FEESchemaRowsetType** complex type.

| Element | Additional constraints | Type definition | Description |
|---|---|---|---|
| Restriction | None | RestrictionElementType | Specifies a restriction on the schema rowset. The enumeration occurs with the result after application of the restriction. |

The following table provides additional information about the attributes of the **FEESchemaRowsetType** type.

| Attribute | Additional constraints | Description |
|---|---|---|
| Connection | None | Specifies a string that contains the connection to schema for enumeration. |
| Schema | None | Specifies the GUID of the schema that is desired for the enumeration. |

### 2.5.1.1.1.4.1 RestrictionElementType

The **RestrictionElementType** complex type specifies a restriction that is applied against a schema.

The following is the XSD of the **RestrictionElementType** complex type.

```xsd
<xs:complexType name="RestrictionElementType">
  <xs:sequence/>
  <xs:attribute name="Index" type="xs:int" use="required" form="unqualified"/>
  <xs:attribute name="Value" type="xs:string" use="required" form="unqualified"/>
</xs:complexType>
```

The **RestrictionElementType** complex type contains no elements.

The following table provides additional information about the attributes of the **RestrictionElementType** complex type.

| Attribute | Additional constraints | Description |
|---|---|---|
| Index | None | An integer that represents the index number of this restriction. |
| Value | None | A string that is the value for this restriction. |

### 2.5.1.1.1.5   FEEFVEType

The **FEEFVEType** complex type specifies enumeration over values that are contained in a variable.

The following is the XSD for the **FEEFVEType** complex type.

```
<xs:complexType name="FEEFVEType">
  <xs:sequence/>
  <xs:attribute name="VariableName" type="xs:string" form="unqualified" use="required"/>
</xs:complexType>
```

The **FEEFVEType** complex type contains no elements.

The following table provides additional information about the attributes for the **FEEFVEType** complex type.

| Attribute | Description |
|---|---|
| VariableName | A string that specifies the name of the variable that contains the information over which the **For…Each** enumerator is performed. |

### 2.5.1.1.1.6 FEENODELISTType

The **FEENODELISTType** complex type specifies enumeration over a list of XML nodes that is obtained by applying an XPATH expression to an XML file.

The following is the XSD for the **FEENODELISTType** complex type.

```
<xs:complexType name="FEENODELISTType">
  <xs:sequence/>
  <xs:attribute name="EnumerationType" type="DTS:NodelistEnumerationTypeEnum"
form="unqualified" use="required"/>
  <xs:attribute name="OuterXPathSourceType" form="unqualified" use="required"
type="DTS:SourceTypeEnum"/>
  <xs:attribute name="OuterXPathString" type="xs:string" form="unqualified"
use="required"/>
  <xs:attribute name="InnerElementType"
type="DTS:NodelistInnerElementEnumerationTypeEnum" use="required" form="unqualified"/>
  <xs:attribute name="InnerXPathSourceType" form="unqualified" use="required"
type="DTS:SourceTypeEnum"/>
  <xs:attribute name="InnerXPathString" type="xs:string" form="unqualified"
use="required"/>
  <xs:attribute name="SourceType" form="unqualified" use="required"
type="DTS:SourceTypeEnum"/>
  <xs:attribute name="SourceDocument" type="xs:string" use="required"
form="unqualified"/>
</xs:complexType>
```

The **FEENODELISTType** complex type contains no elements.

The following table provides additional information about the attributes for the **FEENODELISTType** complex type.

| Attribute | Description |
|---|---|
| EnumerationType | Specifies the type of enumeration. |
| OuterXPathSourceType | Specifies the source type for the XPath [XPATH] string. |
| OuterXPathString | Specifies a string that contains an XPATH expression that is applied to an XML file to |

| Attribute | Description |
|---|---|
| | obtain the nodes over which enumeration occurs. |
| InnerElementType | Specifies the type of enumeration for the inner element. The **InnerElementType** applies only if the **EnumerationType** attribute is set to "ElementCollection". |
| InnerXPathSourceType | Specifies the source type for the inner XPath string. |
| InnerXPathString | Specifies a string that contains an inner XPATH expression that is applied to an XML file to obtain the nodes over which enumeration occurs. |
| SourceType | Specifies the source of the XML. |
| SourceDocument | Specifies a string that contains the name of the source document for the XML. |

### 2.5.1.1.1.6.1 NodelistEnumerationTypeEnum

The **NodelistEnumerationTypeEnum** type contains enumeration values that are used to set the type the enumeration over nodes.

The following is the XSD for the **NodelistEnumerationTypeEnum** type.

```
<xs:simpleType name="NodelistEnumerationTypeEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Navigator"/>
    <xs:enumeration value="Node"/>
    <xs:enumeration value="NodeText"/>
    <xs:enumeration value="ElementCollection"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **NodelistEnumerationTypeEnum** type.

| Enumeration value | Description |
|---|---|
| Navigator | Enumerate using an XPathNavigator. |
| Node | Enumerate nodes that are returned by an XPath operation. |
| NodeText | Enumerate text nodes that are returned by an XPath operation. |
| ElementCollection | Enumerate element nodes that are returned by an XPath operation. |

### 2.5.1.1.1.6.2 NodelistInnerElementEnumerationTypeEnum

The **NodelistInnerElementEnumerationTypeEnum** type contains enumeration values that are used to set the type the enumeration over inner element nodes.

The following is the XSD for the **NodelistInnerElementEnumerationTypeEnum** type.

```
<xs:simpleType name="NodelistInnerElementEnumerationTypeEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Navigator"/>
    <xs:enumeration value="Node"/>
    <xs:enumeration value="NodeText"/>
  </xs:restriction>
```

```
        </xs:simpleType>
```

The following table specifies the enumeration values for the
**NodelistInnerElementEnumerationTypeEnum** type.

| Enumeration value | Description |
|---|---|
| Navigator | Enumerate using an XPathNavigator. |
| Node | Enumerate nodes that are returned by an XPath operation. |
| NodeText | Enumerate text nodes that are returned by an XPath operation. |

#### 2.5.1.1.1.7   FEESMOType

The **FEESMOType** complex type derives enumeration over an SMO object.

The following is the XSD for the **FEESMOType** complex type.

```
<xs:complexType name="FEESMOType">
  <xs:sequence/>
  <xs:attribute name="EnumURN" type="xs:string"
                form="unqualified" use="required"/>
</xs:complexType>
```

The **FEESMOType** complex type contains no elements.

The following table provides additional information about the attributes for the **FEESMOType** complex
type.

| Attribute | Description |
|---|---|
| EnumURN | A string that specifies the content of the SMO object to enumerate. |

### 2.5.1.2  ForEachVariableMappingsType

The **ForEachVariableMappingsType** complex type is the container type for a collection of elements
of type ForEachVariableMappingType.

The following is the XSD of the **ForEachVariableMappingsType** complex type.

```
<xs:complexType name="ForEachVariableMappingsType">
  <xs:sequence>
    <xs:element name="ForEachVariableMapping" type="DTS:ForEachVariableMappingType"
 minOccurs="0" maxOccurs="unbounded" form="qualified"/>
  </xs:sequence>
</xs:complexType>
```

The following table provides additional information about the elements, types, and constraints for the
**ForEachVariableMappingsType** complex type.

| Element | Type definition | Description |
|---|---|---|
| ForEachVariableMapping | ForEachVariableMappingType | Specifies an element of type **ForEachVariableMappingType** that contains information about the variable mapping for a **For...Each** loop and that applies to the element that contains the collection. |

## 2.5.1.2.1 ForEachVariableMappingType

The **ForEachVariableMappingType** complex type MUST NOT be used for an executable unless its **ExecutableType** attribute value is equal to "STOCK:FOREACHLOOP". It contains information about the variable mapping for a **For...Each** enumeration.

The following is the XSD for the **ForEachVariableMappingType** complex type.

```
<xs:complexType name="ForEachVariableMappingType" >
  <xs:sequence>
    <xs:element name="PropertyExpression"
                type="DTS:PropertyExpressionElementType"
                minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
  <xs:attributeGroup ref="DTS:ForEachVariableMappingAttributeGroup"/>
</xs:complexType>
```

The following table provides additional information about the elements, types, and constraints in the **ForEachVariableMappingType** complex type.

| Element | Additional constraints | Type definition | Description |
|---|---|---|---|
| PropertyExpression | Additional constraints exist for the **PropertyExpression** element. These constraints MUST be followed. | PropertyExpressionElementType | The **PropertyExpression** element serves the same purpose as specified for the **Property** element except that an expression that is evaluated at run time is contained in the element's value that is stored in the Integration Services file format. |

The following table provides additional information about the attributes, attribute groups, and types for the **ForEachVariableMappingType** complex type.

| Attribute | Description |
|---|---|
| BasePropertyAttributeGroup | An attribute group that specifies attributes that are common to many different types throughout the schema. |
| ForEachVariableMappingAttributeGroup | An attribute group that specifies attributes for the **ForEachVariableMappingType** complex type. |

## 2.5.1.2.1.1    ForEachVariableMappingAttributeGroup

The **ForEachVariableMappingAttributeGroup** attribute group contains the attributes for the ForEachVariableMappingType type.

The following is the XSD for the **ForEachVariableMappingAttributeGroup** attribute group.

```
<xs:attributeGroup name="ForEachVariableMappingAttributeGroup">
  <xs:attribute name="VariableName" type="xs:string" use="required" form="qualified"/>
  <xs:attribute name="ValueIndex" type="xs:int" use="required" form="qualified"/>
</xs:attributeGroup>
```

The following table specifies the attributes for the **ForEachVariableMappingAttributeGroup** attribute group.

| Attribute | Description |
|---|---|
| VariableName | A string value that specifies the name of the variable to be mapped to the **For…Each** loop. |
| ValueIndex | An integer value that expresses the order of the variables. 0 based. |

### 2.5.1.3 AnyNonPackageExecutableAttributeGroup

The **AnyNonPackageExecutableAttributeGroup** attribute group contains the attributes for the AnyNonPackageExecutableType type.

The following is the XSD for the **AnyNonPackageExecutableAttributeGroup** attribute group.

```
<xs:attributeGroup name="AnyNonPackageExecutableAttributeGroup">
  <!--Pipeline & Tasks use these attributes-->
  <xs:attribute name="ExecutionLocation" type="xs:int" default="0" use="optional"
form="qualified"/>
  <xs:attribute name="ExecutionAddress" type="xs:string" default="" use="optional"
form="qualified"/>
  <xs:attribute name="TaskContact" type="xs:string" default="" use="optional"
form="qualified"/>
  <!--ForLoop uses these attributes-->
  <xs:attribute name="InitExpression" type="xs:string" default="" use="optional"
form="qualified"/>
  <xs:attribute name="EvalExpression" type="xs:string" use="optional" form="qualified"/>
  <xs:attribute name="AssignExpression" type="xs:string" default="" use="optional"
form="qualified"/>
  <!--ForLoop and ForEachLoop use this attribute-->
  <xs:attribute name="MaxConcurrent" type="xs:int" default="1" use="optional"
form="qualified"/>
  <!--Sequence uses this attribute-->
  <xs:attribute name="LockName" type="xs:string" default="" use="optional"
form="qualified"/>
</xs:attributeGroup>
```

The following table specifies the attributes for the **AnyNonPackageExecutableAttributeGroup** attribute group.

| Attribute | Description |
|---|---|
| ExecutionLocation | An integer value. This value SHOULD<22> be set to 0. |
| ExecutionAddress | A string value. This value SHOULD<23> be set to EMPTY. |
| TaskContact | A string value that holds information about the contact point for this package. Intended to be |

| Attribute | Description |
|---|---|
| | helpful in support of the package. This attribute cannot be used for any executable of type **AnyNonPackageExecutableType**. The use of this value depends on the value of the attribute on the **ExecutableType** Executable element. For more information about which **ExecutableType** executables cannot have this attribute value specified, see section 2.6. |
| InitExpression | A string value that specifies an expression to initialize a loop. This attribute cannot be used for any executable of type **AnyNonPackageExecutableType**. The use of the **InitExpression** value depends on the value of the attribute on the **ExecutableType Executable** element. For more information about which **ExecutableType** executables cannot have this attribute value specified, see section 2.6. |
| EvalExpression | A string value that specifies an expression to evaluate to determine whether a loop continues to execute. This attribute cannot be used for any executable of type **AnyNonPackageExecutableType**. The use of this value depends on the value of the attribute on the **ExecutableType Executable** element. For more information about which **ExecutableType** executables cannot have this attribute value specified, see section 2.6. |
| AssignExpression | A string value that specifies the expression to evaluate at the end of each iteration of the loop. This attribute cannot be used for any executable of type **AnyNonPackageExecutableType**. The use of the **AssignExpression** value depends on the value of the attribute on the **ExecutableType Executable** element. For more information about which **ExecutableType** executables cannot have this attribute value specified, see section 2.6. |
| MaxConcurrent | An integer value. This value SHOULD<24> be set to 1. |
| LockName | A string value. This value SHOULD<25> be left blank. |

## 2.6   ExecutableType Attribute Values for Different Executables

The value of the **ExecutableType** attribute of the Executable element carries with it important constraints, which are not fully expressed in the XSD code fragments. These additional constraints MUST be followed for each value of the **ExecutableType** attribute. These additional constraints are explained in this section. The important differences that vary by value of the **ExecutableType** attribute are the following:

- Allowed values for the **Name** attribute of the **Property** element differ for different values of the **ExecutableType** attribute. The same limitations apply to the **PropertyExpression** element within the same **ExecutableType** instance.

- The choice that is contained within the complex type definition for ExecutableObjectDataType is also constrained by this value. Not all of the choices in the XSD are available for each **ExecutableType** attribute value.

- Some **ExecutableType** attribute values allow an **Executables** element to be contained within the current **Executable** element, and others do not allow an **Executables** element to be contained within the current **Executable** element.

### 2.6.1   ExecutableType Instance for Pipeline Task

Pipeline Task maps a data flow from an origin to a destination by passing the data flow through a series of components. An Executable element is a Pipeline Task executable if the **ExecutableType** attribute value is one of the following:

- SSIS.Pipeline

- SSIS.Pipeline.3 (for DTSX2 2012/01)

- STOCK:SSIS.Pipeline

- Microsoft.Pipeline (for DTSX2 2014/01)

The Pipeline Task executable is formally specified to be of type AnyNonPackageExecutableType. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type. This executable MUST follow the anonymous type declaration that is contained in this section.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **AnyNonPackageExecutableType** type:

- This **Executable** element MUST NOT contain an **Executables** element.

- This **Executable** element MUST NOT contain a **ForEachEnumerator** element.

- This **Executable** element MUST NOT contain a **ForEachVariableMappings** element.

- In the **AnyNonPackageExecutableType** type, the type of the **ObjectData** element is given as ExecutableObjectDataType. The **ExecutableObjectDataType** type definition contains an **xs:choice** XSD Schema element. However, for the Pipeline Task executable not all of the choices that are allowed in the **xs:choice** XSD Schema element are available. For this **Executable** element, the **ObjectData** element MUST contain the **pipeline** element of type pipelineObjectDataType.

- The allowed attributes are a restricted subset of those that are allowed on the type, as specified in the AnyNonPackageExecutableAttributeGroup attribute group. Hence, the declaration of the attributes is replaced in this anonymous XSD fragment. The attributes used MUST be restricted to the ones that are shown as valid in the XSD fragment that is contained in this section.

For more information about the elements, attributes, and enumeration values in the following XSD fragment, see section 2.5.1.

```
   <xs:complexType>
     <xs:sequence>
       <xs:element name="ForEachEnumerator" type="DTS:ForEachEnumeratorType" minOccurs="0"
 maxOccurs="0"/>
       <xs:element name="Variables" type="DTS:VariablesType" minOccurs="0"/>
       <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType" />
       <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
 minOccurs="0"/>
       <xs:element name="PrecedenceConstraints" type="DTS:PrecedenceConstraintsType"
 minOccurs="0"/>
       <xs:element name="ForEachVariableMappings" type="DTS:ForEachVariableMappingsType"
 minOccurs="0" maxOccurs="0"/>
       <xs:element name="EventHandlers" type="DTS:EventHandlersType" minOccurs="0"/>
       <xs:element name="ObjectData">
         <xs:complexType>
           <xs:choice>
             <xs:element name="pipeline" type="DTS:pipelineObjectDataType"
 form="unqualified"/>
           </xs:choice>
         </xs:complexType>
       </xs:element>
     </xs:sequence>
     <xs:attribute name="ExecutableType" use="required" type="xs:string"/>
     <xs:attribute name="ThreadHint" use="optional" type="xs:int"/>
     <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
     <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
     <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
     <xs:attribute name="ExecutionLocation" type="xs:int" default="0" use="optional"
 form="qualified"/>
     <xs:attribute name="ExecutionAddress" type="xs:string" default="" use="optional"
 form="qualified"/>
```

```
      <xs:attribute name="TaskContact" type="xs:string" default="" use="optional"
  form="qualified"/>
    </xs:complexType>
```

## 2.6.2 Executable Container Types

Executable container types are not a formal type within the XSD. They are a conceptual subdivision of the values for the **ExecutableType** attribute on the Executable element that is specified by the AnyNonPackageExecutableType type. The **Executable** elements that are specified in the following subsections can contain additional executables that are nested within an instance of themselves.

### 2.6.2.1 ExecutableType Instance for For Each Loop

For Each Loop executable defines a repeating control flow in a package. The loop implementation is similar to a ForEach looping structure in programming languages. An executable is a For Each Loop executable container if the **ExecutableType** attribute value is "STOCK:FOREACHLOOP".

The For Each Loop executable container is formally defined to be of type AnyNonPackageExecutableType. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type than the restrictions that are declared in the **AnyNonPackageExecutableType** type declaration. An Executable element with the **ExecutableType** attribute value equal to "STOCK:FOREACHLOOP" MUST follow the anonymous type declaration that is contained in this section.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **AnyNonPackageExecutableType** type:

▪ This **Executable** element MUST NOT contain an **ObjectData** element.

▪ This **Executable** element MUST NOT contain the **ThreadHint** attribute.

The allowed attributes are a restricted subset of those that are allowed on the type, as specified in the AnyNonPackageExecutableAttributeGroup attribute group. Hence, the declaration of the attributes is replaced in this anonymous XSD fragment. The attributes used MUST be restricted to the ones that are shown as valid in the XSD fragment that is contained in this section.

```
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ForEachEnumerator" type="DTS:ForEachEnumeratorType" minOccurs="1"
maxOccurs="unbounded"/>
      <xs:element name="Variables" type="DTS:VariablesType" minOccurs="0"/>
      <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType" />
      <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
minOccurs="0"/>
      <xs:element name="Executables" type="DTS:ExecutablesType" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element name="PrecedenceConstraints" type="DTS:PrecedenceConstraintsType"
minOccurs="0"/>
      <xs:element name="ForEachVariableMappings" type="DTS:ForEachVariableMappingsType"
minOccurs="1" maxOccurs="unbounded"/>
      <xs:element name="EventHandlers" type="DTS:EventHandlersType" minOccurs="0"/>
      <xs:choice minOccurs="0" maxOccurs="0">
        <xs:element name="ObjectData"/>
      </xs:choice>
    </xs:sequence>
    <xs:attribute name="ExecutableType" use="required" fixed="STOCK:FOREACHLOOP"/>
    <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
    <xs:attribute name="MaxConcurrent" type="xs:int" use="optional" default="1"
form="qualified"/>
  </xs:complexType>
```

### 2.6.2.2 ExecutableType Instance for For Loop

For Loop executable defines a repeating control flow in a package. The loop implementation is similar to a For looping structure in programming languages. An executable is a For Loop executable container if the **ExecutableType** attribute value is "STOCK:FORLOOP".

The For Loop executable container is formally defined to be of type AnyNonPackageExecutableType. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type than the restrictions that are declared in the **AnyNonPackageExecutableType** type declaration. An Executable element with the **ExecutableType** attribute value equal to "STOCK:FORLOOP" MUST follow the anonymous type declaration that is contained in this section.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **AnyNonPackageExecutableType** type:

- This **Executable** element MUST NOT contain an **ObjectData** element.

- This **Executable** element MUST NOT contain a **ForEachEnumerator** element.

- This **Executable** element MUST NOT contain a **ForEachVariableMapping** element.

- This **Executable** element MUST NOT contain the **ThreadHint** attribute.

  The allowed attributes are a restricted subset of those that are allowed on the type, as specified in the AnyNonPackageExecutableAttributeGroup attribute group. Hence, the declaration of the attributes is replaced in this anonymous XSD fragment. The attributes used MUST be restricted to the ones that are shown as valid in the XSD fragment that is contained in this section.

```
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ForEachEnumerator" type="DTS:ForEachEnumeratorType" minOccurs="0"
maxOccurs="0"/>
      <xs:element name="Variables" type="DTS:VariablesType" minOccurs="0"/>
      <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType" />
      <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
minOccurs="0"/>
      <xs:element name="Executables" type="DTS:ExecutablesType" minOccurs="0"
maxOccurs="0"/>
      <xs:element name="PrecedenceConstraints" type="DTS:PrecedenceConstraintsType"
minOccurs="0"/>
      <xs:element name="ForEachVariableMappings" type="DTS:ForEachVariableMappingsType"
minOccurs="0" maxOccurs="0"/>
      <xs:element name="EventHandlers" type="DTS:EventHandlersType" minOccurs="0"/>
      <xs:choice minOccurs="0" maxOccurs="0">
        <xs:element name="ObjectData"/>
      </xs:choice>
    </xs:sequence>
    <xs:attribute name="ExecutableType" use="required" fixed="STOCK:FORLOOP"/>
    <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
    <xs:attribute name="InitExpression" type="xs:string" use="optional" default=""
form="qualified"/>
    <xs:attribute name="EvalExpression" type="xs:string" use="required"
form="qualified"/>
    <xs:attribute name="AssignExpression" type="xs:string" use="optional" default=""
form="qualified"/>
    <xs:attribute name="MaxConcurrent" type="xs:int" use="optional" default="1"
form="qualified"/>
  </xs:complexType>
```

### 2.6.2.3 ExecutableType Instance for Sequence

Sequence executable is a task that serves as a container for other tasks that are executed in a defined sequence. An executable is a Sequence executable container if the **ExecutableType** attribute value is "STOCK:SEQUENCE".

The Sequence executable container is formally specified to be of type AnyNonPackageExecutableType. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type than the restrictions that are declared in the **AnyNonPackageExecutableType** type declaration. An Executable element with the **ExecutableType** attribute value equal to "STOCK:SEQUENCE" MUST follow the anonymous type declaration that is contained in this section.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **AnyNonPackageExecutableType** type:

- This **Executable** element MUST NOT contain an **ObjectData** element.

- This **Executable** element MUST NOT contain a **ForEachEnumerator** element.

- This **Executable** element MUST NOT contain a **ForEachVariableMapping** element.

- This **Executable** element MUST NOT contain the **ThreadHint** attribute.

  The allowed attributes are a restricted subset of those that are allowed on the type, as specified in the AnyNonPackageExecutableAttributeGroup attribute group. Hence, the declaration of the attributes is replaced in this anonymous XSD fragment. The attributes used MUST be restricted to the ones that are shown as valid in the XSD fragment that is contained in this section.

```
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ForEachEnumerator" type="DTS:ForEachEnumeratorType" minOccurs="0"
maxOccurs="0"/>
      <xs:element name="Variables" type="DTS:VariablesType" minOccurs="0"/>
      <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType" />
      <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
minOccurs="0"/>
      <xs:element name="Executables" type="DTS:ExecutablesType" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element name="PrecedenceConstraints" type="DTS:PrecedenceConstraintsType"
minOccurs="0"/>
      <xs:element name="ForEachVariableMappings" type="DTS:ForEachVariableMappingsType"
minOccurs="0" maxOccurs="0"/>
      <xs:element name="EventHandlers" type="DTS:EventHandlersType" minOccurs="0"/>
      <xs:choice minOccurs="0" maxOccurs="0">
        <xs:element name="ObjectData"/>
      </xs:choice>
    </xs:sequence>
    <xs:attribute name="ExecutableType" use="required" fixed="STOCK:SEQUENCE"/>
    <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
    <xs:attribute name="LockName" type="xs:string" use="optional" default=""
form="qualified"/>
  </xs:complexType>
```

### 2.6.3 Executable Task Types

Executable task types are not a formal type within the XSD. They are a conceptual subdivision of the values for the **ExecutableType** attribute on the Executable element that is specified by the AnyNonPackageExecutableType type. The **Executable** elements that are specified in the following subsections MUST NOT contain additional executables nested within an instance of themselves.

### 2.6.3.1 ExecutableType Instance for CDC Control Task

Change Data Capture (CDC) Control Task controls the life cycle of change data capture (CDC) operations. CDC operations record the insert, update, and delete activities that are applied to a Microsoft SQL Server table and then make those change details available in an easily consumed relational format. For more information about CDC, see [MSFT-CDC].

An executable is a Change Data Capture (CDC) Control Task executable if the **ExecutableType** attribute value is one of the following:

▪ Attunity.SqlServer.CDCControlTask.CdcControlTask, Attunity.SqlServer.CDCControlTask, version=1.0.0.0, Culture=neutral, PublicKeyToken=aa342389a732e31c (for DTSX2 2012/01)

▪ Attunity.CdcControlTask (for DTSX2 2014/01)

This executable is formally specified to be of type AnyNonPackageExecutableType. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type. This executable MUST follow the anonymous type declaration that is contained in this section.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the AnyNonPackageExecutableType type:

▪ This **Executable** element MUST NOT contain an **Executables** element.

▪ This **Executable** element MUST NOT contain a **ForEachEnumerator** element.

▪ This **Executable** element MUST NOT contain a **ForEachVariableMappings** element.

▪ In the **AnyNonPackageExecutableType** type, the type of the **ObjectData** element is given as ExecutableObjectDataType. The **ExecutableObjectDataType** type definition contains an **xs:choice** XSD Schema element. However, not all of the choices that are allowed in the **xs:choice** XSD Schema element are available for the CDC Control Task executable. For this executable element, the **ObjectData** element MUST contain the **CdcControlTask** element of type **CdcControlTaskObjectDataType**.

▪ The allowed attributes are a restricted subset of those that are allowed on the type, as specified in the AnyNonPackageExecutableAttributeGroup attribute group. Hence, the declaration of the attributes is replaced in this anonymous XSD fragment. The attributes used MUST be restricted to the ones that are shown as valid in the XSD fragment that is contained in this section.

```
<xs:complexType>
    <xs:sequence>
        <xs:element name="ForEachEnumerator" type="DTS:ForEachEnumeratorType" minOccurs="0"
maxOccurs="0"/>
        <xs:element name="Variables" type="DTS:VariablesType" minOccurs="1"/>
        <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType"/>
        <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
minOccurs="0"/>
        <xs:element name="Executables" type="DTS:ExecutablesType" minOccurs="0"
maxOccurs="0"/>
        <xs:element name="PrecedenceConstraintsType" type="DTS:PrecedenceConstraintsType"
minOccurs="0"/>
        <xs:element name="ForEachVariableMappings" type="DTS:ForEachVariableMappingsType"
minOccurs="0" maxOccurs="0"/>
        <xs:element name="EventHandlers" type="DTS:EventHandlersType" minOccurs="0"/>
        <xs:element name="ObjectData">
            <xs:complexType>
                <xs:choice>
                    <xs:element name="CDCControlTask" form="unqualified"
type="DTS:CDCControlTaskObjectDataType"/>
                </xs:choice>
            </xs:complexType>
```

```
            </xs:element>
        </xs:sequence>
        <xs:attribute name="ExecutableType" use="required" type="xs:string"/>
        <xs:attribute name="ThreadHint" use="optional" type="xs:int"/>
        <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
        <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
        <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
        <xs:attribute name="ExecutionLocation" type="xs:int" default="0" use="optional"
form="qualified"/>
        <xs:attribute name="ExecutionAddress" type="xs:string" default="" use="optional"
form="qualified"/>
        <xs:attribute name="TaskContact" type="xs:string" default="" use="optional"
form="qualified"/>
    </xs:complexType>


    <xs:complexType name="CDCControlTaskObjectDataType">
        <xs:attribute name="ChangeDetectionTimeout" type="xs:int" form="unqualified"
use="optional" default="60"/>
        <xs:attribute name="ChangeDetectionRetryInterval" type="xs:int" form="unqualified"
use="optional" default="10"/>
        <xs:attribute name="CommandTimeout" type="xs:int" form="unqualified" use="optional"
default="30"/>
        <xs:attribute name="StateTable" type="xs:string" use="optional" default=""/>
        <xs:attribute name="StateName" type="xs:string" use="optional" default=""/>
        <xs:attribute name="AutomaticStatePersistence" use="optional" default="True">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:enumeration value="True"/>
                    <xs:enumeration value="False"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="StateVariable" type="xs:string" use="required" default=""/>
        <xs:attribute name="StateConnection" type="xs:string" form="unqualified" use="optional"
default=""/>
        <xs:attribute name="OperationParameter" type="xs:string" form="unqualified"
use="optional" default=""/>
        <xs:attribute name="TaskOperation" use="optional"
type="CDCControlTask:TaskOperationEnum"
                    default="MarkInitialLoadStart"/>
        <xs:attribute name="Connection" type="xs:string" form="unqualified" use="required"/>
    </xs:complexType>


    <xs:simpleType name="TaskOperationEnum">
        <xs:restriction base="xs:string">
            <xs:enumeration value="MarkInitialLoadStart"/>
            <xs:enumeration value="MarkInitialLoadEnd"/>
            <xs:enumeration value="MarkCdcStart"/>
            <xs:enumeration value="GetProcessingRange"/>
            <xs:enumeration value="MarkProcessedRange"/>
        </xs:restriction>
    </xs:simpleType>
```

## 2.6.3.2 ExecutableType Instance for ActiveX Script Task

### *Applies to DTSX2 schema for version 2012/01 (DTSX2 2012/01)*

ActiveX Script Task executes code written as an ActiveX script. An executable is an ActiveX Script Task executable if the **ExecutableType** attribute value is one of the following:

- Microsoft.SqlServer.Dts.Tasks.ActiveXScriptTask.ActiveXScriptTask,
  Microsoft.SqlServer.ActiveXScriptTask, Version=10.0.0.0, Culture=neutral,
  PublicKeyToken=89845dcd8080cc91

▪ STOCK:ActiveXScriptTask

The ActiveX Script Task executable is formally defined to be of type AnyNonPackageExecutableType. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type. This executable MUST follow the anonymous type declaration that is contained in this section.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **AnyNonPackageExecutableType** type:

▪ This **Executable** element MUST NOT contain an **Executable** element.

▪ This **Executable** element MUST NOT contain a **ForEachEnumerator** element.

▪ This **Executable** element MUST NOT contain a **ForEachVariableMapping** element.

▪ In the **AnyNonPackageExecutableType** type, the type of the **ObjectData** element is given as ExecutableObjectDataType. The **ExecutableObjectDataType** type definition contains an **xs:choice** XSD Schema element. However, not all of the choices that are allowed in the **xs:choice** XSD Schema element are available for the ActiveX Script Task executable. For the ActiveX Script Task executable element, the **ObjectData** element MUST contain the ActiveXScriptTaskData element that is specified in the ActiveXScriptTask namespace.

▪ The allowed attributes are a restricted subset of those that are allowed on the type, as specified in the AnyNonPackageExecutableAttributeGroup attribute group. Hence, the declaration of the attributes is replaced in this anonymous XSD fragment. The attributes used MUST be restricted to the ones that are shown as valid in the XSD fragment that is contained in this section.

```
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ForEachEnumerator" type="DTS:ForEachEnumeratorType" minOccurs="0"
maxOccurs="0"/>
      <xs:element name="Variables" type="DTS:VariablesType" minOccurs="0"/>
      <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType" />
      <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
minOccurs="0"/>
      <xs:element name="Executables" type="DTS:ExecutablesType" minOccurs="0"
maxOccurs="0"/>
      <xs:element name="PrecedenceConstraints" type="DTS:PrecedenceConstraintsType"
minOccurs="0"/>
      <xs:element name="ForEachVariableMappings" type="DTS:ForEachVariableMappingsType"
minOccurs="0" maxOccurs="0"/>
      <xs:element name="EventHandlers" type="DTS:EventHandlersType" minOccurs="0"/>
      <xs:element name="ObjectData">
        <xs:complexType>
          <xs:choice>
            <xs:element ref="ActiveXScriptTask:ActiveXScriptTaskData"/>
          </xs:choice>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="ExecutableType" use="required" type="xs:string"/>
    <xs:attribute name="ThreadHint" use="optional" type="xs:int"/>
    <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
    <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
    <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
    <xs:attribute name="ExecutionLocation" type="xs:int" default="0" use="optional"
form="qualified"/>
    <xs:attribute name="ExecutionAddress" type="xs:string" default="" use="optional"
form="qualified"/>
    <xs:attribute name="TaskContact" type="xs:string" default="" use="optional"
form="qualified"/>
  </xs:complexType>
```

### 2.6.3.2.1 TaskExeAttributeGroup

The **TaskExeAttributeGroup** attribute group is used by the anonymous type declarations that are described in the Executable Task Types section. This attribute group is not referenced anywhere in the formal XSD.

The following is the XSD for the **TaskExeAttributeGroup** attribute group.

```
<xs:attributeGroup name="TaskExeAttributeGroup">
   <xs:attribute name="ExecutionLocation" type="xs:int" default="0" use="optional"
form="qualified"/>
   <xs:attribute name="ExecutionAddress" type="xs:string" default="" use="optional"
form="qualified"/>
   <xs:attribute name="TaskContact" type="xs:string" default="" use="optional"
form="qualified"/>
</xs:attributeGroup>
```

The interpretation of the attributes in the **TaskExeAttributeGroup** attribute group is given in AnyNonPackageExecutableAttributeGroup.

### 2.6.3.3 ExecutableType Instance for AS Execute DDL Task

Analysis Services (AS) Execute Data Definition Language (DDL) Task is a task that executes Analysis Services DDL language. AS DDL language is used to create, modify, and manage server objects on an Analysis Services server. An executable is an Analysis Services (AS) Execute Data Definition Language (DDL) Task executable if the **ExecutableType** attribute value is one of the following:

- Microsoft.DataTransformationServices.Tasks.DTSProcessingTask.ASExecuteDDLTask, Microsoft.SqlServer.ASTasks, Version=11.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91 (for DTSX2 2012/01)

- Microsoft.ASExecuteDDLTask (for DTSX2 2014/01)

The AS Execute DDL task executable is formally defined to be of type AnyNonPackageExecutableType. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type. This executable MUST follow the anonymous type declaration that is contained in this section.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **AnyNonPackageExecutableType** type:

- This **Executable** element MUST NOT contain an **Executables** element.

- This **Executable** element MUST NOT contain a **ForEachEnumerator** element.

- This **Executable** element MUST NOT contain a **ForEachVariableMappings** element.

- In the **AnyNonPackageExecutableType** type, the type of the **ObjectData** element is given as ExecutableObjectDataType. The **ExecutableObjectDataType** type definition contains an **xs:choice** XSD Schema element. However, not all of the choices that are allowed in the **xs:choice** XSD Schema element are available for the AS Execute DDL executable. For this **Executable** element, the **ObjectData** element MUST contain the **ASDdlData** element of type ASDdlDataObjectDataType.

- The allowed attributes are a restricted subset of those that are allowed on the type, as specified in the AnyNonPackageExecutableAttributeGroup attribute group. Hence, the declaration of the attributes is replaced in this anonymous XSD fragment. The attributes used MUST be restricted to the ones that are shown as valid in the XSD fragment that is contained in this section.

```
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ForEachEnumerator" type="DTS:ForEachEnumeratorType" minOccurs="0"
  maxOccurs="0"/>
        <xs:element name="Variables" type="DTS:VariablesType" minOccurs="0"/>
        <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType" />
        <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
  minOccurs="0"/>
        <xs:element name="Executables" type="DTS:ExecutablesType" minOccurs="0"
  maxOccurs="0"/>
        <xs:element name="PrecedenceConstraints" type="DTS:PrecedenceConstraintsType"
  minOccurs="0"/>
        <xs:element name="ForEachVariableMappings" type="DTS:ForEachVariableMappingsType"
  minOccurs="0" maxOccurs="0"/>
        <xs:element name="EventHandlers" type="DTS:EventHandlersType" minOccurs="0"/>
        <xs:element name="ObjectData">
          <xs:complexType>
            <xs:choice>
              <xs:element name="ASDdlData" type="DTS:ASDdlDataObjectDataType"/>
            </xs:choice>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attribute name="ExecutableType" use="required" type="xs:string"/>
      <xs:attribute name="ThreadHint" use="optional" type="xs:int"/>
      <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
      <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
      <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
      <xs:attribute name="ExecutionLocation" type="xs:int" default="0" use="optional"
  form="qualified"/>
      <xs:attribute name="ExecutionAddress" type="xs:string" default="" use="optional"
  form="qualified"/>
      <xs:attribute name="TaskContact" type="xs:string" default="" use="optional"
  form="qualified"/>
    </xs:complexType>
```

### 2.6.3.4  ExecutableType Instance for AS Processing Task

Analysis Services (AS) Processing Task is used to initiate the processing of Analysis Services objects by an Analysis Services server. An executable is an AS Processing Task executable if the **ExecutableType** attribute value is one of the following:

- Microsoft.DataTransformationServices.Tasks.DTSProcessingTask.DTSProcessingTask, Microsoft.SqlServer.ASTasks, Version=11.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91 (for DTSX2 2012/01)

- Microsoft.DTSProcessingTask (for DTSX2 2014/01)

The AS Processing Task executable is formally defined to be of type AnyNonPackageExecutableType. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type. This executable MUST follow the anonymous type declaration that is contained in this section.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **AnyNonPackageExecutableType** type:

- This **Executable** element MUST NOT contain an **Executables** element.

- This **Executable** element MUST NOT contain a **ForEachEnumerator** element.

- This **Executable** element MUST NOT contain a **ForEachVariableMappings** element.

- In the **AnyNonPackageExecutableType** type, the type of the **ObjectData** element is given as ExecutableObjectDataType. The **ExecutableObjectDataType** type definition contains an **xs:choice** XSD Schema element. However, not all of the choices that are allowed in the **xs:choice** XSD Schema element are available for the AS Processing Task executable. For this **Executable** element, the **ObjectData** element MUST contain the **ASProcessingData** element of type ASProcessingDataObjectDataType.

- The allowed attributes are a restricted subset of those that are allowed on the type, as specified in the AnyNonPackageExecutableAttributeGroup attribute group. Hence, the declaration of the attributes is replaced in this anonymous XSD fragment. The attributes used MUST be restricted to the ones that are shown as valid in the XSD fragment that is contained in this section.

```
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ForEachEnumerator" type="DTS:ForEachEnumeratorType" minOccurs="0"
maxOccurs="0"/>
      <xs:element name="Variables" type="DTS:VariablesType" minOccurs="0"/>
      <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType" />
      <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
minOccurs="0"/>
      <xs:element name="Executables" type="DTS:ExecutablesType" minOccurs="0"
maxOccurs="0"/>
      <xs:element name="PrecedenceConstraints" type="DTS:PrecedenceConstraintsType"
minOccurs="0"/>
      <xs:element name="ForEachVariableMappings" type="DTS:ForEachVariableMappingsType"
minOccurs="0" maxOccurs="0"/>
      <xs:element name="EventHandlers" type="DTS:EventHandlersType" minOccurs="0"/>
      <xs:element name="ObjectData">
        <xs:complexType>
          <xs:choice>
            <xs:element name="ASProcessingData"
type="DTS:ASProcessingDataObjectDataType"/>
          </xs:choice>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="ExecutableType" use="required" type="xs:string"/>
    <xs:attribute name="ThreadHint" use="optional" type="xs:int"/>
    <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
    <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
    <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
    <xs:attribute name="ExecutionLocation" type="xs:int" default="0" use="optional"
form="qualified"/>
    <xs:attribute name="ExecutionAddress" type="xs:string" default="" use="optional"
form="qualified"/>
    <xs:attribute name="TaskContact" type="xs:string" default="" use="optional"
form="qualified"/>
  </xs:complexType>
```

### 2.6.3.5 ExecutableType Instance for Bulk Insert Task

Bulk Insert Task executes the SQL Server bulk insert statement. An executable is a Bulk Insert Task executable if the **ExecutableType** attribute value is one of the following:

- Microsoft.SqlServer.Dts.Tasks.BulkInsertTask.BulkInsertTask, Microsoft.SqlServer.BulkInsertTask, Version=11.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91 (for DTSX2 2012/01)

- STOCK:BulkInsertTask

- Microsoft.BulkInsertTask (for DTSX2 2014/01)

The Bulk Insert Task executable is formally defined to be of type AnyNonPackageExecutableType. However, the following XSD fragment, which is expressed as an anonymous complex type declaration,

places further restrictions on the type. This executable MUST follow the anonymous type declaration that is contained in this section.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **AnyNonPackageExecutableType** type:

- This **Executable** element MUST NOT contain an **Executables** element.

- This **Executable** element MUST NOT contain a **ForEachEnumerator** element.

- This **Executable** element MUST NOT contain a **ForEachVariableMappings** element.

- In the **AnyNonPackageExecutableType** type, the type of the **ObjectData** element is given as ExecutableObjectDataType. The **ExecutableObjectDataType** type definition contains an **xs:choice** XSD Schema element. However, not all of the choices that are allowed in the **xs:choice** XSD Schema element are available for the Bulk Insert Task executable. For this **Executable** element, the **ObjectData** element MUST contain the BulkInsertTaskData element that is specified in the BulkInsertTask namespace.

- The allowed attributes are a restricted subset of those that are allowed on the type, as specified in the AnyNonPackageExecutableAttributeGroup attribute group. Hence, the declaration of the attributes is replaced in this anonymous XSD fragment. The attributes used MUST be restricted to the ones that are shown as valid in the XSD fragment that is contained in this section.

```
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ForEachEnumerator" type="DTS:ForEachEnumeratorType" minOccurs="0"
maxOccurs="0"/>
      <xs:element name="Variables" type="DTS:VariablesType" minOccurs="0"/>
      <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType" />
      <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
minOccurs="0"/>
      <xs:element name="Executables" type="DTS:ExecutablesType" minOccurs="0"
maxOccurs="0"/>
      <xs:element name="PrecedenceConstraints" type="DTS:PrecedenceConstraintsType"
minOccurs="0"/>
      <xs:element name="ForEachVariableMappings" type="DTS:ForEachVariableMappingsType"
minOccurs="0" maxOccurs="0"/>
      <xs:element name="EventHandlers" type="DTS:EventHandlersType" minOccurs="0"/>
      <xs:element name="ObjectData">
        <xs:complexType>
          <xs:choice>
            <xs:element ref="BulkInsertTask:BulkInsertTaskData"/>
          </xs:choice>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="ExecutableType" use="required" type="xs:string"/>
    <xs:attribute name="ThreadHint" use="optional" type="xs:int"/>
    <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
    <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
    <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
    <xs:attribute name="ExecutionLocation" type="xs:int" default="0" use="optional"
form="qualified"/>
    <xs:attribute name="ExecutionAddress" type="xs:string" default="" use="optional"
form="qualified"/>
    <xs:attribute name="TaskContact" type="xs:string" default="" use="optional"
form="qualified"/>
  </xs:complexType>
```

### 2.6.3.6 ExecutableType Instance for Data Mining Query Task

Data Mining Query Task runs data mining prediction queries based on data mining models that are built in Analysis Services. The prediction query creates a prediction for new data by using mining models. An executable is a Data Mining Query Task executable if the **ExecutableType** attribute value is one of the following:

- Microsoft.SqlServer.Dts.Tasks.DMQueryTask.DMQueryTask, Microsoft.SqlServer.DMQueryTask, Version=11.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91 (for DTSX2 2012/01)

- Microsoft.DMQueryTask (for DTSX2 2014/01)

The Data Mining Query Task executable is formally defined to be of type AnyNonPackageExecutableType. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type. This executable MUST follow the anonymous type declaration that is contained in this section.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **AnyNonPackageExecutableType** type:

- This **Executable** element MUST NOT contain an **Executables** element.

- This **Executable** element MUST NOT contain a **ForEachEnumerator** element.

- This **Executable** element MUST NOT contain a **ForEachVariableMappings** element.

- In the **AnyNonPackageExecutableType** type, the type of the **ObjectData** element is given as ExecutableObjectDataType. The **ExecutableObjectDataType** type definition contains an **xs:choice** XSD Schema element. However, not all of the choices that are allowed in the **xs:choice** XSD Schema element are available for the Data Mining Query Task executable. For this **Executable** element, the **ObjectData** element MUST contain the **DMQueryTaskData** element of type DMQueryTaskDataObjectDataType.

- The allowed attributes are a restricted subset of those that are allowed on the type, as specified in the AnyNonPackageExecutableAttributeGroup attribute group. Hence, the declaration of the attributes is replaced in this anonymous XSD fragment. The attributes used MUST be restricted to the ones that are shown as valid in the XSD fragment that is contained in this section.

```
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ForEachEnumerator" type="DTS:ForEachEnumeratorType" minOccurs="0"
maxOccurs="0"/>
      <xs:element name="Variables" type="DTS:VariablesType" minOccurs="0"/>
      <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType" />
      <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
minOccurs="0"/>
      <xs:element name="Executables" type="DTS:ExecutablesType" minOccurs="0"
maxOccurs="0"/>
      <xs:element name="PrecedenceConstraints" type="DTS:PrecedenceConstraintsType"
minOccurs="0"/>
      <xs:element name="ForEachVariableMappings" type="DTS:ForEachVariableMappingsType"
                  minOccurs="0" maxOccurs="0"/>
      <xs:element name="EventHandlers" type="DTS:EventHandlersType" minOccurs="0"/>
      <xs:element name="ObjectData">
        <xs:complexType>
          <xs:choice>
            <xs:element name="DMQueryTaskData" form="unqualified"
type="DTS:DMQueryTaskDataObjectDataType"/>
          </xs:choice>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="ExecutableType" use="required" type="xs:string"/>
```

```
        <xs:attribute name="ThreadHint" use="optional" type="xs:int"/>
        <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
        <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
        <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
        <xs:attribute name="ExecutionLocation" type="xs:int" default="0" use="optional"
form="qualified"/>
        <xs:attribute name="ExecutionAddress" type="xs:string" default="" use="optional"
form="qualified"/>
        <xs:attribute name="TaskContact" type="xs:string" default="" use="optional"
form="qualified"/>
  </xs:complexType>
```

## 2.6.3.7 ExecutableType Instance for Data Profiling Task

Data Profiling Task computes profiles of data that helps the user understand the values that are present in data and their distribution. An executable is a Data Profiling Task executable if the **ExecutableType** attribute value is one of the following:

▪ Microsoft.SqlServer.Dts.Tasks.DataProfilingTask.DataProfilingTask,
   Microsoft.SqlServer.DataProfilingTask, Version=11.0.0.0, Culture=neutral,
   PublicKeyToken=89845dcd8080cc91 (for DTSX2 2012/01)

▪ Microsoft.DataProfilingTask (for DTSX2 2014/01)

The Data Profiling Task executable is formally defined to be of type AnyNonPackageExecutableType. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type. This executable MUST follow the anonymous type declaration that is contained in this section.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **AnyNonPackageExecutableType** type:

▪ This **Executable** element MUST NOT contain an **Executables** element.

▪ This **Executable** element MUST NOT contain a **ForEachEnumerator** element.

▪ This **Executable** element MUST NOT contain a **ForEachVariableMappings** element.

▪ In the **AnyNonPackageExecutableType** type, the type of the **ObjectData** element is given as ExecutableObjectDataType. The **ExecutableObjectDataType** type definition contains an **xs:choice** XSD Schema element. However, not all of the choices that are allowed in the **xs:choice** XSD Schema element are available for the Data Profiling Task executable. For this **Executable** element, the **ObjectData** element MUST contain the **DataProfilingTaskData** element of type DataProfilingTaskDataObjectDataType.

▪ The allowed attributes are a restricted subset of those that are allowed on the type, as specified in the AnyNonPackageExecutableAttributeGroup attribute group. Hence, the declaration of the attributes is replaced in this anonymous XSD fragment. The attributes used MUST be restricted to the ones that are shown as valid in the XSD fragment that is contained in this section.

```
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ForEachEnumerator" type="DTS:ForEachEnumeratorType" minOccurs="0"
maxOccurs="0"/>
      <xs:element name="Variables" type="DTS:VariablesType" minOccurs="0"/>
      <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType" />
      <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
minOccurs="0"/>
      <xs:element name="Executables" type="DTS:ExecutablesType" minOccurs="0"
maxOccurs="0"/>
      <xs:element name="PrecedenceConstraints" type="DTS:PrecedenceConstraintsType"
minOccurs="0"/>
```

```
            <xs:element name="ForEachVariableMappings" type="DTS:ForEachVariableMappingsType"
    minOccurs="0" maxOccurs="0"/>
            <xs:element name="EventHandlers" type="DTS:EventHandlersType" minOccurs="0"/>
            <xs:element name="ObjectData">
              <xs:complexType>
                <xs:choice>
                  <xs:element name="DataProfilingTaskData" form="unqualified"
    type="DTS:DataProfilingTaskDataObjectDataType"/>
                </xs:choice>
              </xs:complexType>
            </xs:element>
        </xs:sequence>
        <xs:attribute name="ExecutableType" use="required" type="xs:string"/>
        <xs:attribute name="ThreadHint" use="optional" type="xs:int"/>
        <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
        <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
        <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
        <xs:attribute name="ExecutionLocation" type="xs:int" default="0" use="optional"
    form="qualified"/>
        <xs:attribute name="ExecutionAddress" type="xs:string" default="" use="optional"
    form="qualified"/>
        <xs:attribute name="TaskContact" type="xs:string" default="" use="optional"
    form="qualified"/>
      </xs:complexType>
```

## 2.6.3.8 ExecutableType Instance for Execute DTS 2000 Package Task

### *Applies to DTSX2 schema for version 2012/01 (DTSX2 2012/01)*

Data Transformation Services (DTS) 2000 Package is a package file in the format of Microsoft SQL
Server 2000. DTS 2000 Package can execute such packages in Integration Services. Execute DTS
2000 Package Task executes a package from SQL Server 2000. An executable is an Execute DTS 2000
Package Task executable if the **ExecutableType** attribute value is one of the following:

- Microsoft.SqlServer.Dts.Tasks.Exec80PackageTask.Exec80PackageTask,
  Microsoft.SqlServer.Exec80PackageTask, Version=10.0.0.0, Culture=neutral,
  PublicKeyToken=89845dcd8080cc91

- STOCK:Exec80PackageTask

The Execute DTS 2000 Package Task executable is formally defined to be of type
AnyNonPackageExecutableType. However, the following XSD fragment, which is expressed as an
anonymous complex type declaration, places further restrictions on the type. This executable MUST
follow the anonymous type declaration that is contained in this section.

Note the following differences between this anonymous complex type declaration and the full definition
of the complex type for the **AnyNonPackageExecutableType** type:

- This **Executable** element MUST NOT contain an **Executables** element.

- This **Executable** element MUST NOT contain a **ForEachEnumerator** element.

- This **Executable** element MUST NOT contain a **ForEachVariableMappings** element.

- In the **AnyNonPackageExecutableType** type, the type of the **ObjectData** element is given as
  ExecutableObjectDataType. The **ExecutableObjectDataType** type definition contains an
  **xs:choice** XSD Schema element. However, not all of the choices that are allowed in the
  **xs:choice** XSD Schema element are available for the Execute DTS 2000 Package Task executable.
  For this **Executable** element, the **ObjectData** element MUST contain the
  Exec80PackageTaskData element that is specified in the Exec80PackageTask namespace.

- The allowed attributes are a restricted subset of those that are allowed on the type, as specified in
  the AnyNonPackageExecutableAttributeGroup attribute group. Hence, the declaration of the

attributes is replaced in this anonymous XSD fragment. The attributes used MUST be restricted to the ones that are shown as valid in the XSD fragment that is contained in this section.

```xml
<xs:complexType>
  <xs:sequence>
    <xs:element name="ForEachEnumerator" type="DTS:ForEachEnumeratorType" minOccurs="0"
maxOccurs="0"/>
    <xs:element name="Variables" type="DTS:VariablesType" minOccurs="0"/>
    <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType" />
    <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
minOccurs="0"/>
    <xs:element name="Executables" type="DTS:ExecutablesType" minOccurs="0"
maxOccurs="0"/>
    <xs:element name="PrecedenceConstraints" type="DTS:PrecedenceConstraintsType"
minOccurs="0"/>
    <xs:element name="ForEachVariableMappings " type="DTS:ForEachVariableMappingsType"
minOccurs="0" maxOccurs="0"/>
    <xs:element name="EventHandlers" type="DTS:EventHandlersType" minOccurs="0"/>
    <xs:element name="ObjectData">
      <xs:complexType>
        <xs:choice>
          <xs:element ref="Exec80PackageTask:Exec80PackageTaskData"/>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="ExecutableType" use="required" type="xs:string"/>
  <xs:attribute name="ThreadHint" use="optional" type="xs:int"/>
  <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
  <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
  <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
  <xs:attribute name="ExecutionLocation" type="xs:int" default="0" use="optional"
form="qualified"/>
  <xs:attribute name="ExecutionAddress" type="xs:string" default="" use="optional"
form="qualified"/>
  <xs:attribute name="TaskContact" type="xs:string" default="" use="optional"
form="qualified"/>
</xs:complexType>
```

### 2.6.3.9 ExecutableType Instance for Execute Package Task

Execute Package Task executes a package that is stored independently, such as in a file or in a database, of the package that is currently executing. An executable is an Execute Package Task executable if the **ExecutableType** attribute value is one of the following:

▪ SSIS.ExecutePackageTask

▪ SSIS.ExecutePackageTask.3 (for DTSX2 2012/01)

▪ STOCK:ExecutePackageTask

▪ Microsoft.ExecutePackageTask (for DTSX2 2014/01)

The Execute Package Task executable is formally defined to be of type AnyNonPackageExecutableType. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type. This executable MUST follow the anonymous type declaration that is contained in this section.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **AnyNonPackageExecutableType** type:

▪ This **Executable** element MUST NOT contain an **Executables** element.

▪ This **Executable** element MUST NOT contain a **ForEachEnumerator** element.

- This **Executable** element MUST NOT contain a **ForEachVariableMappings** element.

- In the **AnyNonPackageExecutableType** type, the type of the **ObjectData** element is given as ExecutableObjectDataType. The **ExecutableObjectDataType** type definition contains an **xs:choice** XSD Schema element. However, not all of the choices that are allowed in the **xs:choice** XSD Schema element are available for the Execute Package Task executable. For this **Executable** element, the **ObjectData** element MUST contain the **ExecutePackageTask** element of type ExecutePackageTaskObjectDataType.

- The allowed attributes are a restricted subset of those that are allowed on the type, as specified in the AnyNonPackageExecutableAttributeGroup attribute group. Hence, the declaration of the attributes is replaced in this anonymous XSD fragment. The attributes used MUST be restricted to the ones that are shown as valid in the XSD fragment that is contained in this section.

```
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ForEachEnumerator" type="DTS:ForEachEnumeratorType" minOccurs="0"
maxOccurs="0"/>
      <xs:element name="Variables" type="DTS:VariablesType" minOccurs="0"/>
      <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType" />
      <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
minOccurs="0"/>
      <xs:element name="Executables" type="DTS:ExecutablesType" minOccurs="0"
maxOccurs="0"/>
      <xs:element name="PrecedenceConstraints" type="DTS:PrecedenceConstraintsType"
minOccurs="0"/>
      <xs:element name="ForEachVariableMappings" type="DTS:ForEachVariableMappingsType"
                minOccurs="0" maxOccurs="0"/>
      <xs:element name="EventHandlers" type="DTS:EventHandlersType" minOccurs="0"/>
      <xs:element name="ObjectData">
        <xs:complexType>
          <xs:choice>
            <xs:element name="ExecutePackageTask"  form="unqualified"
type="DTS:ExecutePackageTaskObjectDataType"/>
          </xs:choice>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="ExecutableType" use="required" type="xs:string"/>
    <xs:attribute name="ThreadHint" use="optional" type="xs:int"/>
    <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
    <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
    <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
    <xs:attribute name="ExecutionLocation" type="xs:int" default="0" use="optional"
form="qualified"/>
    <xs:attribute name="ExecutionAddress" type="xs:string" default="" use="optional"
form="qualified"/>
    <xs:attribute name="TaskContact" type="xs:string" default="" use="optional"
form="qualified"/>
  </xs:complexType>
```

### 2.6.3.10    ExecutableType Instance for Execute Process Task

Execute Process Task runs an application or batch file as part of a package workflow. An executable is an Execute Process Task executable if the **ExecutableType** attribute value is one of the following:

- Microsoft.SqlServer.Dts.Tasks.ExecuteProcess.ExecuteProcess, Microsoft.SqlServer.ExecProcTask, Version=11.0.0.0, Culture=neutral, , PublicKeyToken=89845dcd8080cc91 (for DTSX2 2012/01)

- STOCK:ExecuteProcessTask

- Microsoft.ExecuteProcess" (for DTSX2 2014/01)

The Execute Process Task executable is formally defined to be of type AnyNonPackageExecutableType. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type. This executable MUST follow the anonymous type declaration that is contained in this section.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **AnyNonPackageExecutableType** type:

- This **Executable** element MUST NOT contain an **Executables** element.

- This **Executable** element MUST NOT contain a **ForEachEnumerator** element.

- This **Executable** element MUST NOT contain a **ForEachVariableMappings** element.

- In the **AnyNonPackageExecutableType** type, the type of the **ObjectData** element is given as ExecutableObjectDataType. The **ExecutableObjectDataType** type definition contains an **xs:choice** XSD Schema element. However, not all of the choices that are allowed in the **xs:choice** XSD Schema element are available for the Execute Process Task executable. For this **Executable** element, the **ObjectData** element MUST contain the **ExecuteProcessData** element of type ExecuteProcessDataObjectDataType.

- The allowed attributes are a restricted subset of those that are allowed on the type, as specified in the AnyNonPackageExecutableAttributeGroup attribute group. Hence, the declaration of the attributes is replaced in this anonymous XSD fragment. The attributes used MUST be restricted to the ones that are shown as valid in the XSD fragment that is contained in this section.

```
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ForEachEnumerator" type="DTS:ForEachEnumeratorType" minOccurs="0"
maxOccurs="0"/>
      <xs:element name="Variables" type="DTS:VariablesType" minOccurs="0"/>
      <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType" />
      <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
minOccurs="0"/>
      <xs:element name="Executables" type="DTS:ExecutablesType" minOccurs="0"
maxOccurs="0"/>
      <xs:element name="PrecedenceConstraints" type="DTS:PrecedenceConstraintsType"
minOccurs="0"/>
      <xs:element name="ForEachVariableMappings" type="DTS:ForEachVariableMappingsType"
minOccurs="0" maxOccurs="0"/>
      <xs:element name="EventHandlers" type="DTS:EventHandlersType" minOccurs="0"/>
      <xs:element name="ObjectData">
        <xs:complexType>
          <xs:choice>
            <xs:element name="ExecuteProcessData" form="unqualified"
type="DTS:ExecuteProcessDataObjectDataType"/>
          </xs:choice>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="ExecutableType" use="required" type="xs:string"/>
    <xs:attribute name="ThreadHint" use="optional" type="xs:int"/>
    <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
    <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
    <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
    <xs:attribute name="ExecutionLocation" type="xs:int" default="0" use="optional"
form="qualified"/>
    <xs:attribute name="ExecutionAddress" type="xs:string" default="" use="optional"
form="qualified"/>
    <xs:attribute name="TaskContact" type="xs:string" default="" use="optional"
form="qualified"/>
  </xs:complexType>
```

### 2.6.3.11 ExecutableType Instance for Execute SQL Task

Execute SQL Task runs SQL statements or stored procedures from a package. An executable is an Execute SQL Task executable if the **ExecutableType** attribute value is one of the following:

- Microsoft.SqlServer.Dts.Tasks.ExecuteSQLTask.ExecuteSQLTask, Microsoft.SqlServer.SQLTask, Version=11.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91 (for DTSX2 2012/01)

- STOCK:SQLTask

- Microsoft.ExecuteSQLTask (for DTSX2 2014/01)

The Execute SQL Task executable is formally defined to be of type AnyNonPackageExecutableType. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type. This executable MUST follow the anonymous type declaration that is contained in this section.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **AnyNonPackageExecutableType** type:

- This **Executable** element MUST NOT contain an **Executables** element.

- This **Executable** element MUST NOT contain a **ForEachEnumerator** element.

- This **Executable** element MUST NOT contain a **ForEachVariableMappings** element.

- In the **AnyNonPackageExecutableType** type, the type of the **ObjectData** element is given as ExecutableObjectDataType. The **ExecutableObjectDataType** type definition contains an **xs:choice** XSD Schema element. However, not all of the choices that are allowed in the **xs:choice** XSD Schema element are available for the Execute SQL Task executable.

  For this **Executable** element, the **ObjectData** element MUST contain the SqlTaskData element that is specified in the SQLTask namespace. The **SqlTaskData** element declaration in the **SQLTask** namespace MUST be replaced by the declaration and type that is specified in this section. This instance of the **SqlTaskData** element MUST contain zero elements and MUST NOT contain attributes other than those that are specified in the SqlTaskBaseAttributeGroup attribute group.

- The allowed attributes are a restricted subset of those that are allowed on the type, as specified in the AnyNonPackageExecutableAttributeGroup attribute group. Hence, the declaration of the attributes is replaced in this anonymous XSD fragment. The attributes used MUST be restricted to the ones that are shown as valid in the XSD fragment that is contained in this section.

```
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ForEachEnumerator" type="DTS:ForEachEnumeratorType" minOccurs="0"
maxOccurs="0"/>
      <xs:element name="Variables" type="DTS:VariablesType" minOccurs="0"/>
      <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType" />
      <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
minOccurs="0"/>
      <xs:element name="Executables" type="DTS:ExecutablesType" minOccurs="0"
maxOccurs="0"/>
      <xs:element name="PrecedenceConstraints" type="DTS:PrecedenceConstraintsType"
minOccurs="0"/>
      <xs:element name="ForEachVariableMappings" type="DTS:ForEachVariableMappingsType"
minOccurs="0" maxOccurs="0"/>
      <xs:element name="EventHandlers" type="DTS:EventHandlersType" minOccurs="0"/>
      <xs:element name="ObjectData">
        <xs:complexType>
          <xs:choice>
            <xs:element ref="SQLTask:SqlTaskData"/>
          </xs:choice>
```

```
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="ExecutableType" use="required" type="xs:string"/>
    <xs:attribute name="ThreadHint" use="optional" type="xs:int"/>
    <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
    <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
    <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
    <xs:attribute name="ExecutionLocation" type="xs:int" default="0" use="optional"
form="qualified"/>
    <xs:attribute name="ExecutionAddress" type="xs:string" default="" use="optional"
form="qualified"/>
    <xs:attribute name="TaskContact" type="xs:string" default="" use="optional"
form="qualified"/>
  </xs:complexType>
```

The preceding anonymous complex type definition requires that the declaration for the **SqlTaskData** element and its type, in the **SQLTask** namespace, be changed to the following.

```
<xs:element name="SqlTaskData" type="SQLTask:SqlTaskDataElementSqlTaskType"/>
<xs:complexType name="SqlTaskDataElementSqlTaskType">
  <xs:sequence/>
  <xs:attributeGroup ref="SQLTask:SqlTaskBaseAttributeGroup"/>
</xs:complexType>
```

### 2.6.3.12      ExecutableType Instance for Expression Task

Expression Task evaluates an expression and assigns the result to a variable. An executable is an Expression Task executable if the **ExecutableType** attribute value is one of the following:

- Microsoft.SqlServer.Dts.Tasks.ExpressionTask.ExpressionTask, Microsoft.SqlServer.ExpressionTask, Version=11.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91 (for DTSX2 2012/01)

- Microsoft.ExpressionTask (for DTSX2 2014/01)

The Expression Task executable is formally specified to be of type AnyNonPackageExecutableType. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type. This executable MUST follow the anonymous type declaration that is contained in this section.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **AnyNonPackageExecutableType** type:

- This **Executable** element MUST NOT contain an **Executables** element.

- This **Executable** element MUST NOT contain a **ForEachEnumerator** element.

- This **Executable** element MUST NOT contain a **ForEachVariableMappings** element.

- In the **AnyNonPackageExecutableType** type, the type of the **ObjectData** element is given as ExecutableObjectDataType. The**ExecutableObjectDataType** type definition contains an **xs:choice** XSD Schema element. However, not all of the choices that are allowed in the **xs:choice** XSD Schema element are available for the Expression Task executable. For this executable element, the **ObjectData** element MUST contain the **ExpressionTask** element of type **ExpressionTaskObjectDataType**.

- The allowed attributes are a restricted subset of those that are allowed on the type, as specified in the AnyNonPackageExecutableAttributeGroup attribute group. Hence, the declaration of the

attributes is replaced in this anonymous XSD fragment. The attributes used MUST be restricted to the ones that are shown as valid in the XSD fragment that is contained in this section.

```
<xs:complexType>
    <xs:sequence>
        <xs:element name="ForEachEnumerator" type="DTS:ForEachEnumeratorType"
minOccurs="0" maxOccurs="0"/>
        <xs:element name="Variables" type="DTS:VariablesType" minOccurs="0"/>
        <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType"/>
        <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
minOccurs="0"/>
        <xs:element name="Executables" type="DTS:ExecutablesType" minOccurs="0"
maxOccurs="0"/>
        <xs:element name="PrecedenceConstraintsType" type="DTS:PrecedenceConstraintsType"
minOccurs="0"/>
        <xs:element name="ForEachVariableMappings" type="DTS:ForEachVariableMappingsType"
minOccurs="0" maxOccurs="0"/>
        <xs:element name="EventHandlers" type="DTS:EventHandlersType" minOccurs="0"/>
        <xs:element name="ObjectData">
            <xs:complexType>
                <xs:choice>
                    <xs:element name="ExpressionTask" form="unqualified"
type="DTS:ExpressionTaskObjectDataType"/>
                </xs:choice>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
    <xs:attribute name="ExecutableType" use="required" type="xs:string"/>
    <xs:attribute name="ThreadHint" use="optional" type="xs:int"/>
    <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
    <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
    <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
    <xs:attribute name="ExecutionLocation" type="xs:int" default="0" use="optional"
form="qualified"/>
    <xs:attribute name="ExecutionAddress" type="xs:string" default="" use="optional"
form="qualified"/>
    <xs:attribute name="TaskContact" type="xs:string" default="" use="optional"
form="qualified"/>
</xs:complexType>


<xs:complexType name="ExpressionTaskObjectDataType">
    <xs:attribute name="Expression" type="xs:string" form="unqualified" use="required"/>
</xs:complexType>
```

### 2.6.3.13    ExecutableType Instance for File System Task

File System Task performs operations on files and directories in the file system. An executable is a File System Task executable if the **ExecutableType** attribute value is one of the following:

- Microsoft.SqlServer.Dts.Tasks.FileSystemTask.FileSystemTask,
  Microsoft.SqlServer.FileSystemTask, Version=11.0.0.0, Culture=neutral,
  PublicKeyToken=89845dcd8080cc91 (for DTSX2 2012/01)

- STOCK:FileSystemTask

- Microsoft.FileSystemTask (for DTSX2 2014/01)

The File System Task executable is formally defined to be of type AnyNonPackageExecutableType. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type. This executable MUST follow the anonymous type declaration that is contained in this section.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **AnyNonPackageExecutableType** type:

- This **Executable** element MUST NOT contain an **Executables** element.

- This **Executable** element MUST NOT contain a **ForEachEnumerator** element.

- This **Executable** element MUST NOT contain a **ForEachVariableMappings** element.

- In the **AnyNonPackageExecutableType** type, the type of the **ObjectData** element is given as ExecutableObjectDataType. The **ExecutableObjectDataType** type definition contains an **xs:choice** XSD Schema element. However, not all of the choices that are allowed in the **xs:choice** XSD Schema element are available for the File System Task executable. For this **Executable** element, the **ObjectData** element MUST contain the **FileSystemData** element of type FileSystemDataObjectDataType.

- The allowed attributes are a restricted subset of those that are allowed on the type, as specified in the AnyNonPackageExecutableAttributeGroup attribute group. Hence, the declaration of the attributes is replaced in this anonymous XSD fragment. The attributes used MUST be restricted to the ones that are shown as valid in the XSD fragment that is contained in this section.

```xml
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ForEachEnumerator" type="DTS:ForEachEnumeratorType" minOccurs="0"
maxOccurs="0"/>
      <xs:element name="Variables" type="DTS:VariablesType" minOccurs="0"/>
      <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType" />
      <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
minOccurs="0"/>
      <xs:element name="Executables" type="DTS:ExecutablesType" minOccurs="0"
maxOccurs="0"/>
      <xs:element name="PrecedenceConstraints" type="DTS:PrecedenceConstraintsType"
minOccurs="0"/>
      <xs:element name="ForEachVariableMappings" type="DTS:ForEachVariableMappingsType"
minOccurs="0" maxOccurs="0"/>
      <xs:element name="EventHandlers" type="DTS:EventHandlersType" minOccurs="0"/>
      <xs:element name="ObjectData">
        <xs:complexType>
          <xs:choice>
            <xs:element name="FileSystemData" form="unqualified"
type="DTS:FileSystemDataObjectDataType"/>
          </xs:choice>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="ExecutableType" use="required" type="xs:string"/>
    <xs:attribute name="ThreadHint" use="optional" type="xs:int"/>
    <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
    <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
    <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
    <xs:attribute name="ExecutionLocation" type="xs:int" default="0" use="optional"
form="qualified"/>
    <xs:attribute name="ExecutionAddress" type="xs:string" default="" use="optional"
form="qualified"/>
    <xs:attribute name="TaskContact" type="xs:string" default="" use="optional"
form="qualified"/>
  </xs:complexType>
```

### 2.6.3.14    ExecutableType Instance for FTP Task

FTP Task uses FTP calls to access, download, upload, and manage files on remote servers. An executable is an FTP Task executable if the **ExecutableType** attribute value is one of the following:

- Microsoft.SqlServer.Dts.Tasks.FtpTask.FtpTask, Microsoft.SqlServer.FtpTask, Version=11.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91" (for DTSX2 2012/01)

- STOCK:FTPTask

- Microsoft:FtpTask (for DTSX2 2014/01)

The FTP Task executable is formally defined to be of type AnyNonPackageExecutableType. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type. This executable MUST follow the anonymous type declaration that is contained in this section.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **AnyNonPackageExecutableType** type:

- This **Executable** element MUST NOT contain an **Executables** element.

- This **Executable** element MUST NOT contain a **ForEachEnumerator** element.

- This **Executable** element MUST NOT contain a **ForEachVariableMappings** element.

- In the **AnyNonPackageExecutableType** type, the type of the **ObjectData** element is given as ExecutableObjectDataType. The **ExecutableObjectDataType** type definition contains an **xs:choice** XSD Schema element. However, not all of the choices that are allowed in the **xs:choice** XSD Schema element are available for the FTP Task executable. For this **Executable** element, the **ObjectData** element MUST contain the **FtpData** element of type FtpDataObjectDataType.

- The allowed attributes are a restricted subset of those that are allowed on the type, as specified in the AnyNonPackageExecutableAttributeGroup attribute group. Hence, the declaration of the attributes is replaced in this anonymous XSD fragment. The attributes used MUST be restricted to the ones that are shown as valid in the XSD fragment that is contained in this section.

```
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ForEachEnumerator" type="DTS:ForEachEnumeratorType" minOccurs="0"
maxOccurs="0"/>
      <xs:element name="Variables" type="DTS:VariablesType" minOccurs="0"/>
      <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType" />
      <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
minOccurs="0"/>
      <xs:element name="Executables" type="DTS:ExecutablesType" minOccurs="0"
maxOccurs="0"/>
      <xs:element name="PrecedenceConstraints" type="DTS:PrecedenceConstraintsType"
minOccurs="0"/>
      <xs:element name="ForEachVariableMappings" type="DTS:ForEachVariableMappingsType"
minOccurs="0" maxOccurs="0"/>
      <xs:element name="EventHandlers" type="DTS:EventHandlersType"
                  minOccurs="0"/>
      <xs:element name="ObjectData">
        <xs:complexType>
          <xs:choice>
            <xs:element name="FtpData" type="DTS:FtpDataObjectDataType"
form="unqualified"/>
          </xs:choice>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="ExecutableType" use="required" type="xs:string"/>
    <xs:attribute name="ThreadHint" use="optional" type="xs:int"/>
    <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
    <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
    <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
```

```
    <xs:attribute name="ExecutionLocation" type="xs:int" default="0" use="optional"
form="qualified"/>
    <xs:attribute name="ExecutionAddress" type="xs:string" default="" use="optional"
form="qualified"/>
    <xs:attribute name="TaskContact" type="xs:string" default="" use="optional"
form="qualified"/>
  </xs:complexType>
```

### 2.6.3.15    ExecutableType Instance for Message Queue Task

Message Queue Task accesses data in Microsoft Message Queuing (MSMQ). An executable is a
Message Queue Task executable if the **ExecutableType** attribute value is one of the following:

- Microsoft.SqlServer.Dts.Tasks.MessageQueueTask.MessageQueueTask,
  Microsoft.SqlServer.MSMQTask, Version=11.0.0.0, Culture=neutral,
  PublicKeyToken=89845dcd8080cc91 (for DTSX2 2012/01)

- STOCK:MSMQTask

- Microsoft:MessageQueueTask (for DTSX2 2014/01)

The Message Queue Task executable is formally defined to be of type AnyNonPackageExecutableType.
However, the following XSD fragment, which is expressed as an anonymous complex type declaration,
places further restrictions on the type. This executable MUST follow the anonymous type declaration
that is contained in this section.

Note the following differences between this anonymous complex type declaration and the full definition
of the complex type for the **AnyNonPackageExecutableType** type:

- This **Executable** element MUST NOT contain an **Executables** element.

- This **Executable** element MUST NOT contain a **ForEachEnumerator** element.

- This **Executable** element MUST NOT contain a **ForEachVariableMappings** element.

- In the **AnyNonPackageExecutableType** type, the type of the **ObjectData** element is given as
  ExecutableObjectDataType. The **ExecutableObjectDataType** type definition contains an
  **xs:choice** XSD Schema element. However, not all of the choices that are allowed in the
  **xs:choice** XSD Schema element are available for the Message Queue Task executable. For this
  **Executable** element, the **ObjectData** element MUST contain the MessageQueueTaskData
  element that is specified in the MessageQueueTask namespace.

- The allowed attributes are a restricted subset of those that are allowed on the type, as specified in
  the AnyNonPackageExecutableAttributeGroup attribute group. Hence, the declaration of the
  attributes is replaced in this anonymous XSD fragment. The attributes used MUST be restricted to
  the ones that are shown as valid in the XSD fragment that is contained in this section.

```
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ForEachEnumerator" type="DTS:ForEachEnumeratorType" minOccurs="0"
maxOccurs="0"/>
      <xs:element name="Variables" type="DTS:VariablesType" minOccurs="0"/>
      <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType" />
      <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
minOccurs="0"/>
      <xs:element name="Executables" type="DTS:ExecutablesType" minOccurs="0"
maxOccurs="0"/>
      <xs:element name="PrecedenceConstraints" type="DTS:PrecedenceConstraintsType"
minOccurs="0"/>
      <xs:element name="ForEachVariableMappings" type="DTS:ForEachVariableMappingsType"
minOccurs="0" maxOccurs="0"/>
      <xs:element name="EventHandlers" type="DTS:EventHandlersType" minOccurs="0"/>
```

```
        <xs:element name="ObjectData">
          <xs:complexType>
            <xs:choice>
              <xs:element ref="MessageQueueTask:MessageQueueTaskData"/>
            </xs:choice>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attribute name="ExecutableType" use="required" type="xs:string"/>
      <xs:attribute name="ThreadHint" use="optional" type="xs:int"/>
      <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
      <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
      <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
      <xs:attribute name="ExecutionLocation" type="xs:int" default="0" use="optional"
  form="qualified"/>
      <xs:attribute name="ExecutionAddress" type="xs:string" default="" use="optional"
  form="qualified"/>
      <xs:attribute name="TaskContact" type="xs:string" default="" use="optional"
  form="qualified"/>
    </xs:complexType>
```

### 2.6.3.16    ExecutableType Instance for Script Task

Script Task runs custom code that is created in a programming environment within the data flow. An executable is a Script Task executable if the **ExecutableType** attribute value is one of the following:

▪ Microsoft.SqlServer.Dts.Tasks.ScriptTask.ScriptTask, Microsoft.SqlServer.ScriptTask, Version=11.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91 (for DTSX2 2012/01)

▪ STOCK:ScriptTask

▪ Microsoft:ScriptTask (for DTSX2 2014/01)

The Script Task executable is formally defined to be of type AnyNonPackageExecutableType. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type. This executable MUST follow the anonymous type declaration that is contained in this section.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **AnyNonPackageExecutableType** type:

▪ This **Executable** element MUST NOT contain an **Executables** element.

▪ This **Executable** element MUST NOT contain a **ForEachEnumerator** element.

▪ This **Executable** element MUST NOT contain a **ForEachVariableMappings** element.

▪ In the **AnyNonPackageExecutableType** type, the type of the **ObjectData** element is given as ExecutableObjectDataType. The **ExecutableObjectDataType** type definition contains an **xs:choice** XSD Schema element. However, not all of the choices that are allowed in the **xs:choice** XSD Schema element are available for the Script Task executable. For this **Executable** element, the **ObjectData** element MUST contain the **ScriptProject** element of type ScriptProjectObjectDataType.

▪ The allowed attributes are a restricted subset of those that are allowed on the type, as specified in the AnyNonPackageExecutableAttributeGroup attribute group. Hence, the declaration of the attributes is replaced in this anonymous XSD fragment. The attributes used MUST be restricted to the ones that are shown as valid in the XSD fragment that is contained in this section.

```
   <xs:complexType>
     <xs:sequence>
```

```
        <xs:element name="ForEachEnumerator" type="DTS:ForEachEnumeratorType" minOccurs="0"
    maxOccurs="0"/>
        <xs:element name="Variables" type="DTS:VariablesType" minOccurs="0"/>
        <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType" />
        <xs:element name="PropertyExpression"
                    type="DTS:PropertyExpressionElementType" minOccurs="0"/>
        <xs:element name="Executables" type="DTS:ExecutablesType"
                    minOccurs="0" maxOccurs="0"/>
        <xs:element name="PrecedenceConstraints"
                    type="DTS:PrecedenceConstraintsType" minOccurs="0"/>
        <xs:element name="ForEachVariableMappings" type="DTS:ForEachVariableMappingsType"
    minOccurs="0" maxOccurs="0"/>
        <xs:element name="EventHandlers" type="DTS:EventHandlersType" minOccurs="0"/>
        <xs:element name="ObjectData">
          <xs:complexType>
            <xs:choice>
              <xs:element name="ScriptProject" type="DTS:ScriptProjectObjectDataType"
    form="unqualified"/>
            </xs:choice>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attribute name="ExecutableType" use="required" type="xs:string"/>
      <xs:attribute name="ThreadHint" use="optional" type="xs:int"/>
      <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
      <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
      <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
      <xs:attribute name="ExecutionLocation" type="xs:int" default="0" use="optional"
    form="qualified"/>
      <xs:attribute name="ExecutionAddress" type="xs:string" default="" use="optional"
    form="qualified"/>
      <xs:attribute name="TaskContact" type="xs:string" default="" use="optional"
    form="qualified"/>
    </xs:complexType>
```

### 2.6.3.17 ExecutableType Instance for Send Mail Task

Send Mail Task sends email messages. An executable is a Send Mail Task executable if the **ExecutableType** attribute value is one of the following:

- Microsoft.SqlServer.Dts.Tasks.SendMailTask.SendMailTask, Microsoft.SqlServer.SendMailTask, Version=11.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91 (for DTSX2 2012/01)

- STOCK:SendMailTask

- Microsoft:SendMailTask (for DTSX2 2014/01)

The Send Mail Task executable is formally defined to be of type AnyNonPackageExecutableType. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type. This executable MUST follow the anonymous type declaration that is contained in this section.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **AnyNonPackageExecutableType** type:

- This **Executable** element MUST NOT contain an **Executables** element.

- This **Executable** element MUST NOT contain a **ForEachEnumerator** element.

- This **Executable** element MUST NOT contain a **ForEachVariableMappings** element.

- In the **AnyNonPackageExecutableType** type, the type of the **ObjectData** element is given as ExecutableObjectDataType. The **ExecutableObjectDataType** type definition contains an **xs:choice** XSD Schema element. However, not all of the choices that are allowed in the

**xs:choice** XSD Schema element are available for the Send Mail Task executable. For this **Executable** element, the **ObjectData** element MUST contain the SendMailTaskData element that is specified in the SendMailTask namespace.

- The allowed attributes are a restricted subset of those that are allowed on the type, as specified in the AnyNonPackageExecutableAttributeGroup attribute group. Hence, the declaration of the attributes is replaced in this anonymous XSD fragment. The attributes used MUST be restricted to the ones that are shown as valid in the XSD fragment that is contained in this section.

```
<xs:complexType>
  <xs:sequence>
    <xs:element name="ForEachEnumerator" type="DTS:ForEachEnumeratorType" minOccurs="0"
maxOccurs="0"/>
    <xs:element name="Variables" type="DTS:VariablesType" minOccurs="0"/>
    <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType" />
    <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
minOccurs="0"/>
    <xs:element name="Executables" type="DTS:ExecutablesType" minOccurs="0"
maxOccurs="0"/>
    <xs:element name="PrecedenceConstraints" type="DTS:PrecedenceConstraintsType"
minOccurs="0"/>
    <xs:element name="ForEachVariableMappings" type="DTS:ForEachVariableMappingsType"
minOccurs="0" maxOccurs="0"/>
    <xs:element name="EventHandlers" type="DTS:EventHandlersType" minOccurs="0"/>
    <xs:element name="ObjectData">
      <xs:complexType>
        <xs:choice>
          <xs:element ref="SendMailTask:SendMailTaskData"/>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="ExecutableType" use="required" type="xs:string"/>
  <xs:attribute name="ThreadHint" use="optional" type="xs:int"/>
  <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
  <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
  <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
  <xs:attribute name="ExecutionLocation" type="xs:int" default="0" use="optional"
form="qualified"/>
  <xs:attribute name="ExecutionAddress" type="xs:string" default="" use="optional"
form="qualified"/>
  <xs:attribute name="TaskContact" type="xs:string" default="" use="optional"
form="qualified"/>
</xs:complexType>
```

### 2.6.3.18    ExecutableType Instance for Transfer Database Task

Transfer Database Task copies or moves a database between different instances of the database server. An executable is a Transfer Database Task executable if the **ExecutableType** attribute value is one of the following:

- Microsoft.SqlServer.Dts.Tasks.TransferDatabaseTask.TransferDatabaseTask, Microsoft.SqlServer.TransferDatabasesTask, Version=11.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91 (for DTSX2 2012/01)

- STOCK:TransferDatabaseTask

- Microsoft:TransferDatabaseTask (for DTSX2 2014/01)

The Transfer Database Task executable is formally defined to be of type AnyNonPackageExecutableType. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type. This executable MUST follow the anonymous type declaration that is contained in this section.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **AnyNonPackageExecutableType** type:

- This **Executable** element MUST NOT contain an **Executables** element.

- This **Executable** element MUST NOT contain a **ForEachEnumerator** element.

- This **Executable** element MUST NOT contain a **ForEachVariableMappings** element.

- In the **AnyNonPackageExecutableType** type, the type of the **ObjectData** element is given as ExecutableObjectDataType. The **ExecutableObjectDataType** type definition contains an **xs:choice** XSD Schema element. However, not all of the choices that are allowed in the **xs:choice** XSD Schema element are available for the Transfer Database Task executable. For this **Executable** element, the **ObjectData** element MUST contain the **TransferDatabaseTaskData** element of type TransferDatabaseTaskDataObjectDataType.

- The allowed attributes are a restricted subset of those that are allowed on the type, as specified in the AnyNonPackageExecutableAttributeGroup attribute group. Hence, the declaration of the attributes is replaced in this anonymous XSD fragment. The attributes used MUST be restricted to the ones that are shown as valid in the XSD fragment that is contained in this section.

```
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ForEachEnumerator" type="DTS:ForEachEnumeratorType" minOccurs="0"
maxOccurs="0"/>
      <xs:element name="Variables" type="DTS:VariablesType" minOccurs="0"/>
      <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType" />
      <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
minOccurs="0"/>
      <xs:element name="Executables" type="DTS:ExecutablesType" minOccurs="0"
maxOccurs="0"/>
      <xs:element name="PrecedenceConstraints" type="DTS:PrecedenceConstraintsType"
minOccurs="0"/>
      <xs:element name="ForEachVariableMappings" type="DTS:ForEachVariableMappingsType"
minOccurs="0" maxOccurs="0"/>
      <xs:element name="EventHandlers" type="DTS:EventHandlersType" minOccurs="0"/>
      <xs:element name="ObjectData">
        <xs:complexType>
          <xs:choice>
            <xs:element name="TransferDatabasesTaskData" form="unqualified"
type="DTS:TransferDatabasesTaskDataObjectDataType"/>
          </xs:choice>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="ExecutableType" use="required" type="xs:string"/>
    <xs:attribute name="ThreadHint" use="optional" type="xs:int"/>
    <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
    <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
    <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
    <xs:attribute name="ExecutionLocation" type="xs:int" default="0" use="optional"
form="qualified"/>
    <xs:attribute name="ExecutionAddress" type="xs:string" default="" use="optional"
form="qualified"/>
    <xs:attribute name="TaskContact" type="xs:string" default="" use="optional"
form="qualified"/>
  </xs:complexType>
```

### 2.6.3.19    ExecutableType Instance for Transfer Error Messages Task

Transfer Error Messages Task transfers user-defined error messages between instances of a database. An executable is a Transfer Error Messages Task executable if the **ExecutableType** attribute value is one of the following:

- Microsoft.SqlServer.Dts.Tasks.TransferErrorMessagesTask.TransferErrorMessagesTask, Microsoft.SqlServer.TransferErrorMessagesTask, Version=11.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91 (for DTSX2 2012/01)

- STOCK:TransferErrorMessagesTask

- Microsoft:TransferErrorMessagesTask" (for DTSX2 2014/01)

The Transfer Error Messages Task executable is formally defined to be of type AnyNonPackageExecutableType. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type. This executable MUST follow the anonymous type declaration that is contained in this section.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **AnyNonPackageExecutableType** type:

- This **Executable** element MUST NOT contain an **Executables** element.

- This **Executable** element MUST NOT contain a **ForEachEnumerator** element.

- This **Executable** element MUST NOT contain a **ForEachVariableMappings** element.

- In the **AnyNonPackageExecutableType** type, the type of the **ObjectData** element is given as ExecutableObjectDataType. The **ExecutableObjectDataType** type definition contains an **xs:choice** XSD Schema element. However, not all of the choices that are allowed in the **xs:choice** XSD Schema element are available for the Transfer Error Messages Task executable. For this **Executable** element, the **ObjectData** element MUST contain the **TransferErrorMessagesTaskData** element of type TransferErrorMessagesTaskDataObjectDataType.

- The allowed attributes are a restricted subset of those that are allowed on the type, as specified in the AnyNonPackageExecutableAttributeGroup attribute group. Hence, the declaration of the attributes is replaced in this anonymous XSD fragment. The attributes used MUST be restricted to the ones that are shown as valid in the XSD fragment that is contained in this section.

```
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ForEachEnumerator" type="DTS:ForEachEnumeratorType" minOccurs="0"
maxOccurs="0"/>
      <xs:element name="Variables" type="DTS:VariablesType" minOccurs="0"/>
      <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType" />
      <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
minOccurs="0"/>
      <xs:element name="Executables" type="DTS:ExecutablesType" minOccurs="0"
maxOccurs="0"/>
      <xs:element name="PrecedenceConstraints" type="DTS:PrecedenceConstraintsType"
minOccurs="0"/>
      <xs:element name="ForEachVariableMappings" minOccurs="0" maxOccurs="0"/>
      <xs:element name="EventHandlers" type="DTS:EventHandlersType" minOccurs="0"/>
      <xs:element name="ObjectData">
        <xs:complexType>
          <xs:choice>
            <xs:element name="TransferErrorMessagesTaskData" form="unqualified"
type="DTS:TransferErrorMessagesTaskDataObjectDataType"/>
          </xs:choice>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="ExecutableType" use="required" type="xs:string"/>
    <xs:attribute name="ThreadHint" use="optional" type="xs:int"/>
    <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
    <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
    <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
```

```
        <xs:attribute name="ExecutionLocation" type="xs:int" default="0" use="optional"
form="qualified"/>
        <xs:attribute name="ExecutionAddress" type="xs:string" default="" use="optional"
form="qualified"/>
        <xs:attribute name="TaskContact" type="xs:string" default="" use="optional"
form="qualified"/>
    </xs:complexType>
```

### 2.6.3.20    ExecutableType Instance for Transfer Jobs Task

Transfer Jobs Task transfers SQL Server Agent Jobs between instances of a SQL Server database. An executable is a Transfer Jobs Task executable if the **ExecutableType** attribute value is one of the following:

▪ Microsoft.SqlServer.Dts.Tasks.TransferJobsTask.TransferJobsTask,
  Microsoft.SqlServer.TransferJobsTask, Version=11.0.0.0, Culture=neutral,
  PublicKeyToken=89845dcd8080cc91" (for DTSX2 2012/01)

▪ STOCK:TransferJobsTask

▪ Microsoft:TransferJobsTask (for DTSX2 2014/01)

The Transfer Jobs Task executable is formally defined to be of type AnyNonPackageExecutableType. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type. This executable MUST follow the anonymous type declaration that is contained in this section.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **AnyNonPackageExecutableType** type:

▪ This **Executable** element MUST NOT contain an **Executables** element.

▪ This **Executable** element MUST NOT contain a **ForEachEnumerator** element.

▪ This **Executable** element MUST NOT contain a **ForEachVariableMappings** element.

▪ In the **AnyNonPackageExecutableType** type, the type of the **ObjectData** element is given as ExecutableObjectDataType. The **ExecutableObjectDataType** type definition contains an **xs:choice** XSD Schema element. However, not all of the choices that are allowed in the **xs:choice** XSD Schema element are available for the Transfer Jobs Task executable. For this **Executable** element, the **ObjectData** element MUST contain the **TransferJobsTaskData** element of type TransferJobsTaskDataObjectDataType.

▪ The allowed attributes are a restricted subset of those that are allowed on the type, as specified in the AnyNonPackageExecutableAttributeGroup attribute group. Hence, the declaration of the attributes is replaced in this anonymous XSD fragment. The attributes used MUST be restricted to the ones that are shown as valid in the XSD fragment that is contained in this section.

```
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ForEachEnumerator" type="DTS:ForEachEnumeratorType" minOccurs="0"
maxOccurs="0"/>
        <xs:element name="Variables" type="DTS:VariablesType" minOccurs="0"/>
        <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType" />
        <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
minOccurs="0"/>
        <xs:element name="Executables" type="DTS:ExecutablesType" minOccurs="0"
maxOccurs="0"/>
        <xs:element name="PrecedenceConstraints" type="DTS:PrecedenceConstraintsType"
minOccurs="0"/>
        <xs:element name="ForEachVariableMappings" type="DTS:ForEachVariableMappingsType"
minOccurs="0" maxOccurs="0"/>
```

```
            <xs:element name="EventHandlers" type="DTS:EventHandlersType" minOccurs="0"/>
            <xs:element name="ObjectData">
              <xs:complexType>
                <xs:choice>
                  <xs:element name="TransferJobsTaskData" form="unqualified"
  type="DTS:TransferJobsTaskDataObjectDataType"/>
                </xs:choice>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
          <xs:attribute name="ExecutableType" use="required" type="xs:string"/>
          <xs:attribute name="ThreadHint" use="optional" type="xs:int"/>
          <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
          <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
          <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
          <xs:attribute name="ExecutionLocation" type="xs:int" default="0" use="optional"
  form="qualified"/>
          <xs:attribute name="ExecutionAddress" type="xs:string" default="" use="optional"
  form="qualified"/>
          <xs:attribute name="TaskContact" type="xs:string" default="" use="optional"
  form="qualified"/>
        </xs:complexType>
```

### 2.6.3.21    ExecutableType Instance for Transfer Logins Task

Transfer Logins Task transfers user logins between instances of a database. An executable is a
Transfer Logins Task executable if the **ExecutableType** attribute value is one of the following:

▪   Microsoft.SqlServer.Dts.Tasks.TransferLoginsTask.TransferLoginsTask,
    Microsoft.SqlServer.TransferLoginsTask, Version=11.0.0.0, Culture=neutral,
    PublicKeyToken=89845dcd8080cc91 (for DTSX2 2012/01)

▪   STOCK:TransferLoginsTask

▪   Microsoft:TransferLoginsTask (for DTSX2 2014/01)

The Transfer Logins Task executable is formally defined to be of type AnyNonPackageExecutableType.
However, the following XSD fragment, which is expressed as an anonymous complex type declaration,
places further restrictions on the type. This executable MUST follow the anonymous type declaration
that is contained in this section.

Note the following differences between this anonymous complex type declaration and the full definition
of the complex type for the **AnyNonPackageExecutableType** type:

▪   This **Executable** element MUST NOT contain an **Executables** element.

▪   This **Executable** element MUST NOT contain a **ForEachEnumerator** element.

▪   This **Executable** element MUST NOT contain a **ForEachVariableMappings** element.

▪   In the **AnyNonPackageExecutableType** type, the type of the **ObjectData** element is given as
    ExecutableObjectDataType. The **ExecutableObjectDataType** type definition contains an
    **xs:choice** XSD Schema element. However, not all of the choices that are allowed in the
    **xs:choice** XSD Schema element are available for the Transfer Logins Task executable. For this
    **Executable** element, the **ObjectData** element MUST contain the **TransferLoginsTaskData**
    element of type TransferLoginsTaskDataObjectDataType.

▪   The allowed attributes are a restricted subset of those that are allowed on the type, as specified in
    the AnyNonPackageExecutableAttributeGroup attribute group. Hence, the declaration of the
    attributes is replaced in this anonymous XSD fragment. The attributes used MUST be restricted to
    the ones that are shown as valid in the XSD fragment that is contained in this section.

```
        <xs:complexType>
          <xs:sequence>
            <xs:element name="ForEachEnumerator" type="DTS:ForEachEnumeratorType" minOccurs="0"
maxOccurs="0"/>
            <xs:element name="Variables" type="DTS:VariablesType" minOccurs="0"/>
            <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType" />
            <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
minOccurs="0"/>
            <xs:element name="Executables" type="DTS:ExecutablesType" minOccurs="0"
maxOccurs="0"/>
            <xs:element name="PrecedenceConstraints" type="DTS:PrecedenceConstraintsType"
minOccurs="0"/>
            <xs:element name="ForEachVariableMappings" type="DTS:ForEachVariableMappingsType"
minOccurs="0" maxOccurs="0"/>
            <xs:element name="EventHandlers" type="DTS:EventHandlersType" minOccurs="0"/>
            <xs:element name="ObjectData">
              <xs:complexType>
                <xs:choice>
                  <xs:element name="TransferLoginsTaskData" form="unqualified"
type="DTS:TransferLoginsTaskDataObjectDataType"/>
                </xs:choice>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
          <xs:attribute name="ExecutableType" use="required" type="xs:string"/>
          <xs:attribute name="ThreadHint" use="optional" type="xs:int"/>
          <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
          <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
          <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
          <xs:attribute name="ExecutionLocation" type="xs:int" default="0" use="optional"
form="qualified"/>
          <xs:attribute name="ExecutionAddress" type="xs:string" default="" use="optional"
form="qualified"/>
          <xs:attribute name="TaskContact" type="xs:string" default="" use="optional"
form="qualified"/>
        </xs:complexType>
```

### 2.6.3.22    ExecutableType Instance for Transfer SQL Server Objects Task

Transfer SQL Server Objects Task transfers objects between instances of SQL Server. An executable is a Transfer SQL Server Objects Task executable if the **ExecutableType** attribute value is one of the following:

- Microsoft.SqlServer.Dts.Tasks.TransferSqlServerObjectsTask.TransferSqlServerObjectsTask, Microsoft.SqlServer.TransferSqlServerObjectsTask, Version=11.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91 (for DTSX2 2012/01)

- STOCK:TransferSqlServerObjectsTask

- Microsoft:TransferSQLServerObjectsTask (for DTSX2 2014/01)

The Transfer SQL Server Objects Task executable is formally defined to be of type AnyNonPackageExecutableType. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type. This executable MUST follow the anonymous type declaration that is contained in this section.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **AnyNonPackageExecutableType** type:

- This **Executable** element MUST NOT contain an **Executables** element.

- This **Executable** element MUST NOT contain a **ForEachEnumerator** element.

- This **Executable** element MUST NOT contain a **ForEachVariableMappings** element.

- In the **AnyNonPackageExecutableType** type, the type of the **ObjectData** element is given as ExecutableObjectDataType. The **ExecutableObjectDataType** type definition contains an **xs:choice** XSD Schema element. However, not all of the choices that are allowed in the **xs:choice** XSD Schema element are available for the Transfer SQL Server Objects Task executable. For this **Executable** element, the **ObjectData** element MUST contain the **TransferSqlServerObjectsTaskData** element of type TransferSqlServerObjectsTaskDataObjectDataType.

- The allowed attributes are a restricted subset of those that are allowed on the type, as specified in the AnyNonPackageExecutableAttributeGroup attribute group. Hence, the declaration of the attributes is replaced in this anonymous XSD fragment. The attributes used MUST be restricted to the ones that are shown as valid in the XSD fragment that is contained in this section.

```
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ForEachEnumerator" type="DTS:ForEachEnumeratorType" minOccurs="0"
maxOccurs="0"/>
      <xs:element name="Variables" type="DTS:VariablesType" minOccurs="0"/>
      <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType" />
      <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
minOccurs="0"/>
      <xs:element name="Executables" type="DTS:ExecutablesType" minOccurs="0"
maxOccurs="0"/>
      <xs:element name="PrecedenceConstraints" type="DTS:PrecedenceConstraintsType"
minOccurs="0"/>
      <xs:element name="ForEachVariableMappings" type="DTS:ForEachVariableMappingsType"
minOccurs="0" maxOccurs="0"/>
      <xs:element name="EventHandlers" type="DTS:EventHandlersType" minOccurs="0"/>
      <xs:element name="ObjectData">
        <xs:complexType>
          <xs:choice>
            <xs:element name="TransferSqlServerObjectsTaskData" form="unqualified"
type="DTS:TransferSqlServerObjectsTaskDataObjectDataType"/>
          </xs:choice>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="ExecutableType" use="required" type="xs:string"/>
    <xs:attribute name="ThreadHint" use="optional" type="xs:int"/>
    <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
    <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
    <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
    <xs:attribute name="ExecutionLocation" type="xs:int" default="0" use="optional"
form="qualified"/>
    <xs:attribute name="ExecutionAddress" type="xs:string" default="" use="optional"
form="qualified"/>
    <xs:attribute name="TaskContact" type="xs:string" default="" use="optional"
form="qualified"/>
  </xs:complexType>
```

### 2.6.3.23    ExecutableType Instance for Transfer Stored Procedures Task

Transfer Stored Procedures Task transfers stored procedures between database instances. An executable is a Transfer Stored Procedures Task executable if the **ExecutableType** attribute value is one of the following:

- Microsoft.SqlServer.Dts.Tasks.TransferStoredProceduresTask.TransferStoredProceduresTask, Microsoft.SqlServer.TransferStoredProceduresTask, Version=11.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91 (for DTSX2 2012/01)

- STOCK:TransferStoredProceduresTask

- Microsoft:TransferStoredProceduresTask (for DTSX2 2014/01)

The Transfer Stored Procedures Task executable is formally defined to be of type AnyNonPackageExecutableType. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type. This executable MUST follow the anonymous type declaration that is contained in this section.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **AnyNonPackageExecutableType** type:

- This **Executable** element MUST NOT contain an **Executables** element.

- This **Executable** element MUST NOT contain a **ForEachEnumerator** element.

- This **Executable** element MUST NOT contain a **ForEachVariableMappings** element.

- In the **AnyNonPackageExecutableType** type, the type of the **ObjectData** element is given as ExecutableObjectDataType. The **ExecutableObjectDataType** type definition contains an **xs:choice** XSD Schema element. However, not all of the choices that are allowed in the **xs:choice** XSD Schema element are available for the Transfer Stored Procedures Task executable. For this **Executable** element, the **ObjectData** element MUST contain the **TransferStoredProceduresTaskData** element of type TransferStoredProceduresTaskDataObjectDataType.

- The allowed attributes are a restricted subset of those that are allowed on the type, as specified in the AnyNonPackageExecutableAttributeGroup attribute group. Hence, the declaration of the attributes is replaced in this anonymous XSD fragment. The attributes used MUST be restricted to the ones that are shown as valid in the XSD fragment that is contained in this section.

```xml
<xs:complexType>
  <xs:sequence>
    <xs:element name="ForEachEnumerator" type="DTS:ForEachEnumeratorType" minOccurs="0"
maxOccurs="0"/>
    <xs:element name="Variables" type="DTS:VariablesType" minOccurs="0"/>
    <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType" />
    <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
minOccurs="0"/>
    <xs:element name="Executables" type="DTS:ExecutablesType" minOccurs="0"
maxOccurs="0"/>
    <xs:element name="PrecedenceConstraints" type="DTS:PrecedenceConstraintsType"
minOccurs="0"/>
    <xs:element name="ForEachVariableMappings" type="DTS:ForEachVariableMappingsType"
minOccurs="0" maxOccurs="0"/>
    <xs:element name="EventHandlers" type="DTS:EventHandlersType" minOccurs="0"/>
    <xs:element name="ObjectData">
      <xs:complexType>
        <xs:choice>
          <xs:element name="TransferStoredProceduresTaskData" form="unqualified"
type="DTS:TransferStoredProceduresTaskDataObjectDataType"/>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="ExecutableType" use="required" type="xs:string"/>
  <xs:attribute name="ThreadHint" use="optional" type="xs:int"/>
  <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
  <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
  <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
  <xs:attribute name="ExecutionLocation" type="xs:int" default="0" use="optional"
form="qualified"/>
  <xs:attribute name="ExecutionAddress" type="xs:string" default="" use="optional"
form="qualified"/>
  <xs:attribute name="TaskContact" type="xs:string" default="" use="optional"
form="qualified"/>
</xs:complexType>
```

### 2.6.3.24 ExecutableType Instance for Web Service Task

Web Service Task executes a call to a web service method. An executable is a Web Service Task executable if the **ExecutableType** attribute value is one of the following:

▪ Microsoft.SqlServer.Dts.Tasks.WebServiceTask.WebServiceTask, Microsoft.SqlServer.WebServiceTask, Version=11.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91 (for DTSX2 2012/01)

▪ STOCK:WebServiceTask

▪ Microsoft:WebServiceTask (for DTSX2 2014/01)

The Web Service Task executable is formally defined to be of type AnyNonPackageExecutableType. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type. This executable MUST follow the anonymous type declaration that is contained in this section.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **AnyNonPackageExecutableType** type:

▪ This **Executable** element MUST NOT contain an **Executables** element.

▪ This **Executable** element MUST NOT contain a **ForEachEnumerator** element.

▪ This **Executable** element MUST NOT contain a **ForEachVariableMappings** element.

▪ In the **AnyNonPackageExecutableType** type, the type of the **ObjectData** element is given as ExecutableObjectDataType. The **ExecutableObjectDataType** type definition contains an **xs:choice** XSD Schema element. However, not all of the choices that are allowed in the **xs:choice** XSD Schema element are available for the Web Service Task executable. For this **Executable** element, the **ObjectData** element MUST contain the WebServiceTaskData element that is specified in the WebServiceTask namespace.

▪ The allowed attributes are a restricted subset of those that are allowed on the type, as specified in the AnyNonPackageExecutableAttributeGroup attribute group. Hence, the declaration of the attributes is replaced in this anonymous XSD fragment. The attributes used MUST be restricted to the ones that are shown as valid in the XSD fragment that is contained in this section.

```
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ForEachEnumerator" type="DTS:ForEachEnumeratorType" minOccurs="0"
maxOccurs="0"/>
      <xs:element name="Variables" type="DTS:VariablesType" minOccurs="0"/>
      <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType" />
      <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
minOccurs="0"/>
      <xs:element name="Executables" type="DTS:ExecutablesType" minOccurs="0"
maxOccurs="0"/>
      <xs:element name="PrecedenceConstraints" type="DTS:PrecedenceConstraintsType"
minOccurs="0"/>
      <xs:element name="ForEachVariableMappings" type="DTS:ForEachVariableMappingsType"
minOccurs="0" maxOccurs="0"/>
      <xs:element name="EventHandlers" type="DTS:EventHandlersType"
                minOccurs="0"/>
      <xs:element name="ObjectData">
        <xs:complexType>
          <xs:choice>
            <xs:element ref="WSTask:WebServiceTaskData"/>
          </xs:choice>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
```

```
            <xs:attribute name="ExecutableType" use="required" type="xs:string"/>
            <xs:attribute name="ThreadHint" use="optional" type="xs:int"/>
            <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
            <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
            <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
            <xs:attribute name="ExecutionLocation" type="xs:int" default="0" use="optional"
    form="qualified"/>
            <xs:attribute name="ExecutionAddress" type="xs:string" default="" use="optional"
    form="qualified"/>
            <xs:attribute name="TaskContact" type="xs:string" default="" use="optional"
    form="qualified"/>
        </xs:complexType>
```

### 2.6.3.25    ExecutableType Instance for WMI Data Reader Task

Windows Management Instrumentation (WMI) Data Reader Task runs queries by using the WMI Query Language (WQL) that returns information from WMI about a computer system. An executable is a WMI Data Reader Task executable if the **ExecutableType** attribute value is one of the following:

▪ Microsoft.SqlServer.Dts.Tasks.WmiDataReaderTask.WmiDataReaderTask,
   Microsoft.SqlServer.WMIDRTask, Version=11.0.0.0, Culture=neutral,
   PublicKeyToken=89845dcd8080cc91 (for DTSX2 2012/01)

▪ STOCK:WmiDataReaderTask

▪ Microsoft:WmiDataReaderTask (for DTSX2 2014/01)

The WMI Data Reader Task executable is formally defined to be of type AnyNonPackageExecutableType. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type. This executable MUST follow the anonymous type declaration that is contained in this section.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **AnyNonPackageExecutableType** type:

▪ This **Executable** element MUST NOT contain an **Executables** element.

▪ This **Executable** element MUST NOT contain a **ForEachEnumerator** element.

▪ This **Executable** element MUST NOT contain a **ForEachVariableMappings** element.

▪ In the **AnyNonPackageExecutableType** type, the type of the **ObjectData** element is given as ExecutableObjectDataType. The **ExecutableObjectDataType** type definition contains an **xs:choice** XSD Schema element. However, not all of the choices that are allowed in the **xs:choice** XSD Schema element are available for the WMI Data Reader Task executable. For this **Executable** element, the **ObjectData** element MUST contain the **WMIDRTaskData** element of type WMIDRTaskDataObjectDataType.

▪ The allowed attributes are a restricted subset of those that are allowed on the type, as specified in the AnyNonPackageExecutableAttributeGroup attribute group. Hence, the declaration of the attributes is replaced in this anonymous XSD fragment. The attributes used MUST be restricted to the ones that are shown as valid in the XSD fragment that is contained in this section.

```
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ForEachEnumerator" type="DTS:ForEachEnumeratorType" minOccurs="0"
    maxOccurs="0"/>
        <xs:element name="Variables" type="DTS:VariablesType" minOccurs="0"/>
        <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType" />
        <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
    minOccurs="0"/>
```

```
        <xs:element name="Executables" type="DTS:ExecutablesType" minOccurs="0"
maxOccurs="0"/>
        <xs:element name="PrecedenceConstraints" type="DTS:PrecedenceConstraintsType"
minOccurs="0"/>
        <xs:element name="ForEachVariableMappings" type="DTS:ForEachVariableMappingsType"
minOccurs="0" maxOccurs="0"/>
        <xs:element name="EventHandlers" type="DTS:EventHandlersType" minOccurs="0"/>
        <xs:element name="ObjectData">
          <xs:complexType>
            <xs:choice>
              <xs:element name="WMIDRTaskData" type="DTS:WMIDRTaskDataObjectDataType"
form="unqualified"/>
            </xs:choice>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attribute name="ExecutableType" use="required" type="xs:string"/>
      <xs:attribute name="ThreadHint" use="optional" type="xs:int"/>
      <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
      <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
      <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
      <xs:attribute name="ExecutionLocation" type="xs:int" default="0" use="optional"
form="qualified"/>
      <xs:attribute name="ExecutionAddress" type="xs:string" default="" use="optional"
form="qualified"/>
      <xs:attribute name="TaskContact" type="xs:string" default="" use="optional"
form="qualified"/>
  </xs:complexType>
```

### 2.6.3.26    ExecutableType Instance for WMI Event Watcher Task

Windows Management Instrumentation (WMI) Event Watcher Task watches for a WMI event by using a WMI Query Language (WQL) event query to specify events of interest. An executable is a WMI Event Watcher Task executable if the **ExecutableType** attribute value is one of the following:

▪ Microsoft.SqlServer.Dts.Tasks.WmiEventWatcherTask.WmiEventWatcherTask,
  Microsoft.SqlServer.WMIEWTask, Version=11.0.0.0, Culture=neutral,
  PublicKeyToken=89845dcd8080cc91 (for DTSX2 2012/01)

▪ STOCK:WmiEventWatcherTask

▪ Microsoft:WmiEventWatcherTask (for DTSX2 2014/01)

The WMI Event Watcher Task executable is formally defined to be of type AnyNonPackageExecutableType. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type. This executable MUST follow the anonymous type declaration that is contained in this section.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **AnyNonPackageExecutableType** type:

▪ This **Executable** element MUST NOT contain an **Executables** element.

▪ This **Executable** element MUST NOT contain a **ForEachEnumerator** element.

▪ This **Executable** element MUST NOT contain a **ForEachVariableMappings** element.

▪ In the **AnyNonPackageExecutableType** type, the type of the **ObjectData** element is given as ExecutableObjectDataType. The **ExecutableObjectDataType** type definition contains an **xs:choice** XSD Schema element. However, not all of the choices that are allowed in the **xs:choice** XSD Schema element are available for the WMI Event Watcher Task executable. For this **Executable** element, the **ObjectData** element MUST contain the **WMIEWTaskData** element of type WMIEWTaskDataObjectDataType.

- The allowed attributes are a restricted subset of those that are allowed on the type, as specified in the AnyNonPackageExecutableAttributeGroup attribute group. Hence, the declaration of the attributes is replaced in this anonymous XSD fragment. The attributes used MUST be restricted to the ones that are shown as valid in the XSD fragment that is contained in this section.

```
<xs:complexType>
  <xs:sequence>
    <xs:element name="ForEachEnumerator" type="DTS:ForEachEnumeratorType" minOccurs="0"
maxOccurs="0"/>
    <xs:element name="Variables" type="DTS:VariablesType" minOccurs="0"/>
    <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType" />
    <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
minOccurs="0"/>
    <xs:element name="Executables" type="DTS:ExecutablesType" minOccurs="0"
maxOccurs="0"/>
    <xs:element name="PrecedenceConstraints" type="DTS:PrecedenceConstraintsType"
minOccurs="0"/>
    <xs:element name="ForEachVariableMappings" type="DTS:ForEachVariableMappingsType"
minOccurs="0" maxOccurs="0"/>
    <xs:element name="EventHandlers" type="DTS:EventHandlersType" minOccurs="0"/>
    <xs:element name="ObjectData">
      <xs:complexType>
        <xs:choice>
          <xs:element
          name="WMIEWTaskData" type="DTS:WMIEWTaskDataObjectDataType"
form="unqualified"/>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="ExecutableType" use="required" type="xs:string"/>
  <xs:attribute name="ThreadHint" use="optional" type="xs:int"/>
  <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
  <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
  <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
  <xs:attribute name="ExecutionLocation" type="xs:int" default="0" use="optional"
form="qualified"/>
  <xs:attribute name="ExecutionAddress" type="xs:string" default="" use="optional"
form="qualified"/>
  <xs:attribute name="TaskContact" type="xs:string" default="" use="optional"
form="qualified"/>
</xs:complexType>
```

### 2.6.3.27    ExecutableType Instance for XML Task

XML Task is used to work with XML data. By using this task, a package can retrieve XML documents; apply operations to the documents by using Extensible Stylesheet Language Transformations (XSLT) style sheets and XPath expressions; merge multiple documents; and validate, compare, and save the updated documents to files and variables. An executable is an XML Task executable if the **ExecutableType** attribute value is one of the following:

- Microsoft.SqlServer.Dts.Tasks.XMLTask.XMLTask, Microsoft.SqlServer.XMLTask, Version=11.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91 (for DTSX2 2012/01)

- STOCK:XMLTask

- Microsoft:XMLTask (for DTSX2 2014/01)

The XML Task executable is formally defined to be of type AnyNonPackageExecutableType. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type. This executable MUST follow the anonymous type declaration that is contained in this section.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **AnyNonPackageExecutableType** type:

- This **Executable** element MUST NOT contain an **Executables** element.

- This **Executable** element MUST NOT contain a **ForEachEnumerator** element.

- This **Executable** element MUST NOT contain a **ForEachVariableMappings** element.

- In the **AnyNonPackageExecutableType** type, the type of the **ObjectData** element is given as ExecutableObjectDataType. The **ExecutableObjectDataType** type definition contains an **xs:choice** XSD Schema element. However, not all of the choices that are allowed in the **xs:choice** XSD Schema element are available for the XML Task executable. For this **Executable** element, the **ObjectData** element MUST contain the **XMLTaskData** element of type XMLTaskDataObjectDataType.

- The allowed attributes are a restricted subset of those that are allowed on the type, as specified in the AnyNonPackageExecutableAttributeGroup attribute group. Hence, the declaration of the attributes is replaced in this anonymous XSD fragment. The attributes used MUST be restricted to the ones that are shown as valid in the XSD fragment that is contained in this section.

```xml
<xs:complexType>
  <xs:sequence>
    <xs:element name="ForEachEnumerator" type="DTS:ForEachEnumeratorType"
minOccurs="0" maxOccurs="0"/>
    <xs:element name="Variables" type="DTS:VariablesType" minOccurs="0"/>
    <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType" />
    <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
minOccurs="0"/>
    <xs:element name="Executables" type="DTS:ExecutablesType" minOccurs="0"
maxOccurs="0"/>
    <xs:element name="PrecedenceConstraints" type="DTS:PrecedenceConstraintsType"
minOccurs="0"/>
    <xs:element name="ForEachVariableMappings"
type="DTS:ForEachVariableMappingsType" minOccurs="0" maxOccurs="0"/>
    <xs:element name="EventHandlers" type="DTS:EventHandlersType" minOccurs="0"/>
    <xs:element name="ObjectData">
      <xs:complexType>
        <xs:choice>
          <xs:element name="XMLTaskData" type="DTS:XMLTaskDataObjectDataType"
form="unqualified"/>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="ExecutableType" use="required" type="xs:string"/>
  <xs:attribute name="ThreadHint" use="optional" type="xs:int"/>
  <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
  <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
  <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
  <xs:attribute name="ExecutionLocation" type="xs:int" default="0" use="optional"
form="qualified"/>
  <xs:attribute name="ExecutionAddress" type="xs:string" default="" use="optional"
form="qualified"/>
  <xs:attribute name="TaskContact" type="xs:string" default="" use="optional"
form="qualified"/>
</xs:complexType>
```

### 2.6.4 Executable Maintenance Tasks

Executable maintenance task types are not a formal type within the XSD. They are a conceptual subdivision of the values for the **ExecutableType** attribute on the Executable element, as specified by

the AnyNonPackageExecutableType type. The **Executable** elements that are specified in the following subsections MUST NOT contain additional executables nested within an instance of themselves.

### 2.6.4.1 ExecutableType Instance for Backup Task

Backup Task is used to add a backup task to a database maintenance plan. An executable is a Backup Task executable if the **ExecutableType** attribute value is one of the following:

▪ Microsoft.SqlServer.Management.DatabaseMaintenance.DbMaintenanceBackupTask, Microsoft.SqlServer.MaintenancePlanTasks, Version=11.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91 (for DTSX2 2012/01)

▪ Microsoft.DbMaintenanceBackupTask (for DTSX2 2014/01)

The Backup Task executable is formally defined to be of type AnyNonPackageExecutableType. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type. This executable MUST follow the anonymous type declaration that is contained in this section.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **AnyNonPackageExecutableType** type:

▪ This **Executable** element MUST NOT contain an **Executables** element.

▪ This **Executable** element MUST NOT contain a **ForEachEnumerator** element.

▪ This **Executable** element MUST NOT contain a **ForEachVariableMappings** element.

▪ In the **AnyNonPackageExecutableType** type, the type of the **ObjectData** element is given as ExecutableObjectDataType. The **ExecutableObjectDataType** type definition contains an **xs:choice** XSD Schema element. However, all of the choices that are allowed in the **xs:choice** XSD Schema element are not available for the Backup Task executable.

For this **Executable** element, the **ObjectData** element MUST contain the SqlTaskData element that is specified in the SQLTask namespace. The **SqlTaskData** element declaration in the **SQLTask** namespace MUST be replaced by the declaration and type that is specified in this section. This instance of the **SqlTaskData** element MUST NOT contain elements except for the **SelectedDatabases** element and the **BackupDestinationList** element. This instance of the **SqlTaskData** element MUST NOT contain attributes other than those that are specified in the SqlTaskBaseAttributeGroup attribute group, the MaintenanceTaskBaseAttributeGroup attribute group, the BackupTaskAttributeGroup attribute group, and the **DatabaseSelectionType** attribute.

▪ The allowed attributes are a restricted subset of those that are allowed on the type, as specified in the AnyNonPackageExecutableAttributeGroup attribute group. Hence, the declaration of the attributes is replaced in this anonymous XSD fragment. The attributes used MUST be restricted to the ones that are shown as valid in the XSD fragment that is contained in this section.

```
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ForEachEnumerator" type="DTS:ForEachEnumeratorType" minOccurs="0"
maxOccurs="0"/>
        <xs:element name="Variables" type="DTS:VariablesType" minOccurs="0"/>
        <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType" />
        <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
minOccurs="0"/>
        <xs:element name="Executables" type="DTS:ExecutablesType" minOccurs="0"
maxOccurs="0"/>
        <xs:element name="PrecedenceConstraints" type="DTS:PrecedenceConstraintsType"
minOccurs="0"/>
```

```
            <xs:element name="ForEachVariableMappings" type="DTS:ForEachVariableMappingsType"
    minOccurs="0" maxOccurs="0"/>
            <xs:element name="EventHandlers" type="DTS:EventHandlersType" minOccurs="0"/>
            <xs:element name="ObjectData">
              <xs:complexType>
                <xs:choice>
                  <xs:element ref="SQLTask:SqlTaskData"/>
                </xs:choice>
              </xs:complexType>
            </xs:element>
        </xs:sequence>
        <xs:attribute name="ExecutableType" use="required" type="xs:string"/>
        <xs:attribute name="ThreadHint" use="optional" type="xs:int"/>
        <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
        <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
        <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
        <xs:attribute name="ExecutionLocation" type="xs:int" default="0" use="optional"
    form="qualified"/>
        <xs:attribute name="ExecutionAddress" type="xs:string" default="" use="optional"
    form="qualified"/>
        <xs:attribute name="TaskContact" type="xs:string" default="" use="optional"
    form="qualified"/>
      </xs:complexType>
```

The preceding anonymous complex type definition requires that the declaration for the **SqlTaskData** element and its type, in the **SQLTask** namespace, be changed to the following.

```
    <xs:element name="SqlTaskData" type="SQLTask:SqlTaskDataElementBackupTaskType"/>
    <xs:complexType name="SqlTaskDataElementCheckIntegrityTaskType">
      <xs:sequence>
        <xs:element name="SelectedDatabases" type="SQLTask:SelectedDatabasesType"
    minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="BackupDestinationList" type="SQLTask:BackupDestinationListType"
    minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attributeGroup ref="SQLTask:SqlTaskBaseAttributeGroup"/>
      <xs:attributeGroup ref="SQLTask:MaintenanceTaskBaseAttributeGroup"/>
      <xs:attributeGroup  ref="SQLTask:CheckIntegrityTaskAttributeGroup"/>
      <xs:attribute name="DatabaseSelectionType"
    type="SQLTask:SqlTaskDatabaseSelectionTypeEnum"/>
    </xs:complexType>
```

## 2.6.4.2 ExecutableType Instance for Check Integrity Task

Check Integrity Task checks the allocation and structural integrity of all the objects in a database. An executable is a Check Integrity Task executable if the **ExecutableType** attribute value is one of the following:

▪ Microsoft.SqlServer.Management.DatabaseMaintenance.DbMaintenanceCheckIntegrityTask, Microsoft.SqlServer.MaintenancePlanTasks, Version=11.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91 (for DTSX2 2012/01)

▪ Microsoft.DbMaintenanceCheckIntegrityTask (for DTSX2 2014/01)

The Check Integrity Task executable is formally defined to be of type AnyNonPackageExecutableType. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type. This executable MUST follow the anonymous type declaration that is contained in this section.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **AnyNonPackageExecutableType** type:

▪ This **Executable** element MUST NOT contain an **Executables** element.

- This **Executable** element MUST NOT contain a **ForEachEnumerator** element.

- This **Executable** element MUST NOT contain a **ForEachVariableMappings** element.

- In the **AnyNonPackageExecutableType** type, the type of the **ObjectData** element is given as ExecutableObjectDataType. The **ExecutableObjectDataType** type definition contains an **xs:choice** XSD Schema element. However, all of the choices that are allowed in the **xs:choice** XSD Schema element are not available for the Check Integrity Task executable.

  For this **Executable** element, the **ObjectData** element MUST contain the SqlTaskData element that is specified in the SQLTask namespace. The **SqlTaskData** element declaration in the **SQLTask** namespace MUST be replaced by the declaration and type that is specified in this section. This instance of the **SqlTaskData** element MUST NOT contain elements except for the **SelectedDatabases** element. This instance of the **SqlTaskData** element MUST NOT contain attributes other than those that are specified in the SqlTaskBaseAttributeGroup attribute group, the MaintenanceTaskBaseAttributeGroup attribute group, the CheckIntegrityTaskAttributeGroup attribute group, and the **DatabaseSelectionType** attribute.

- The allowed attributes are a restricted subset of those that are allowed on the type, as specified in the AnyNonPackageExecutableAttributeGroup attribute group. Hence, the declaration of the attributes is replaced in this anonymous XSD fragment. The attributes used MUST be restricted to the ones that are shown as valid in the XSD fragment that is contained in this section.

```
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ForEachEnumerator" type="DTS:ForEachEnumeratorType" minOccurs="0"
maxOccurs="0"/>
      <xs:element name="Variables" type="DTS:VariablesType" minOccurs="0"/>
      <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType" />
      <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
minOccurs="0"/>
      <xs:element name="Executables" type="DTS:ExecutablesType" minOccurs="0"
maxOccurs="0"/>
      <xs:element name="PrecedenceConstraints" type="DTS:PrecedenceConstraintsType"
minOccurs="0"/>
      <xs:element name="ForEachVariableMappings" type="DTS:ForEachVariableMappingsType"
minOccurs="0" maxOccurs="0"/>
      <xs:element name="EventHandlers" type="DTS:EventHandlersType" minOccurs="0"/>
      <xs:element name="ObjectData">
        <xs:complexType>
          <xs:choice>
            <xs:element ref="SQLTask:SqlTaskData"/>
          </xs:choice>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="ExecutableType" use="required" type="xs:string"/>
    <xs:attribute name="ThreadHint" use="optional" type="xs:int"/>
    <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
    <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
    <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
    <xs:attribute name="ExecutionLocation" type="xs:int" default="0" use="optional"
form="qualified"/>
    <xs:attribute name="ExecutionAddress" type="xs:string" default="" use="optional"
form="qualified"/>
    <xs:attribute name="TaskContact" type="xs:string" default="" use="optional"
form="qualified"/>
  </xs:complexType>
```

The preceding anonymous complex type definition requires that the declaration for the **SqlTaskData** element and its type, in the **SQLTask** namespace, be changed to the following.

```
  <xs:element name="SqlTaskData"
type="SQLTask:SqlTaskDataElementCheckIntegrityTaskType"/>
    <xs:complexType name="SqlTaskDataElementCheckIntegrityTaskType">
      <xs:sequence>
        <xs:element name="SelectedDatabases" type="SQLTask:SelectedDatabasesType"
minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attributeGroup ref="SQLTask:SqlTaskBaseAttributeGroup"/>
      <xs:attributeGroup ref="SQLTask:MaintenanceTaskBaseAttributeGroup"/>
      <xs:attributeGroup ref="SQLTask:CheckIntegrityTaskAttributeGroup"/>
      <xs:attribute name="DatabaseSelectionType"
type="SQLTask:SqlTaskDatabaseSelectionTypeEnum"/>
    </xs:complexType>
```

### 2.6.4.3 ExecutableType Instance for Execute SQL Agent Job Task

Execute SQL Agent Job Task runs SQL Server Agent jobs. SQL Server Agent is a Microsoft Windows service that runs jobs that have been defined in an instance of SQL Server. An executable is an Execute SQL Agent Job Task executable if the **ExecutableType** attribute value is one of the following:

▪ Microsoft.SqlServer.Management.DatabaseMaintenance.DbMaintenanceExecuteAgentJobTask, Microsoft.SqlServer.MaintenancePlanTasks, Version=11.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91 (for DTSX2 2012/01)

▪ Microsoft.DbMaintenanceExecuteAgentJobTask (for DTSX2 2014/01)

The Execute SQL Agent Job Task executable is formally defined to be of type AnyNonPackageExecutableType. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type. This executable MUST follow the anonymous type declaration that is contained in this section.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **AnyNonPackageExecutableType** type:

▪ This **Executable** element MUST NOT contain an **Executables** element.

▪ This **Executable** element MUST NOT contain a **ForEachEnumerator** element.

▪ This **Executable** element MUST NOT contain a **ForEachVariableMappings** element.

▪ In the **AnyNonPackageExecutableType** type, the type of the **ObjectData** element is given as ExecutableObjectDataType. The **ExecutableObjectDataType** type definition contains an **xs:choice** XSD Schema element. However, all of the choices that are allowed in the **xs:choice** XSD Schema element are not available for the Execute SQL Agent Job Task executable.

For this **Executable** element, the **ObjectData** element MUST contain the SqlTaskData element that is specified in the SQLTask namespace. The **SqlTaskData** element declaration in the **SQLTask** namespace MUST be replaced by the declaration and type that is specified in this section. This instance of the **SqlTaskData** element MUST contain zero elements. This instance of the **SqlTaskData** element MUST NOT contain attributes other than those that are specified in the SqlTaskBaseAttributeGroup attribute group, the MaintenanceTaskBaseAttributeGroup attribute group, and the AgentJobTaskAttributeGroup attribute group.

▪ The allowed attributes are a restricted subset of those that are allowed on the type, as specified in the AnyNonPackageExecutableAttributeGroup attribute group. Hence, the declaration of the attributes is replaced in this anonymous XSD fragment. The attributes used MUST be restricted to the ones that are shown as valid in the XSD fragment that is contained in this section.

```
  <xs:complexType>
    <xs:sequence>
```

```
        <xs:element name="ForEachEnumerator" type="DTS:ForEachEnumeratorType" minOccurs="0"
maxOccurs="0"/>
        <xs:element name="Variables" type="DTS:VariablesType" minOccurs="0"/>
        <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType" />
        <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
minOccurs="0"/>
        <xs:element name="Executables" type="DTS:ExecutablesType" minOccurs="0"
maxOccurs="0"/>
        <xs:element name="PrecedenceConstraints" type="DTS:PrecedenceConstraintsType"
minOccurs="0"/>
        <xs:element name="ForEachVariableMappings" type="DTS:ForEachVariableMappingsType"
minOccurs="0" maxOccurs="0"/>
        <xs:element name="EventHandlers" type="DTS:EventHandlersType" minOccurs="0"/>
        <xs:element name="ObjectData">
          <xs:complexType>
            <xs:choice>
              <xs:element ref="SQLTask:SqlTaskData"/>
            </xs:choice>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attribute name="ExecutableType" use="required" type="xs:string"/>
      <xs:attribute name="ThreadHint" use="optional" type="xs:int"/>
      <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
      <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
      <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
      <xs:attribute name="ExecutionLocation" type="xs:int" default="0" use="optional"
form="qualified"/>
      <xs:attribute name="ExecutionAddress" type="xs:string" default="" use="optional"
form="qualified"/>
      <xs:attribute name="TaskContact" type="xs:string" default="" use="optional"
form="qualified"/>
    </xs:complexType>
```

The preceding anonymous complex type definition requires that the declaration for the **SqlTaskData** element and its type, in the **SQLTask** namespace, be changed to the following.

```
<xs:element name="SqlTaskData" type="SQLTask:SqlTaskDataElementAgentJobTaskType"/>
<xs:complexType name="SqlTaskDataElementAgentJobTaskType">
  <xs:sequence/>
  <xs:attributeGroup ref="SQLTask:SqlTaskBaseAttributeGroup"/>
  <xs:attributeGroup ref="SQLTask:MaintenanceTaskBaseAttributeGroup"/>
  <xs:attributeGroup ref="SQLTask:AgentJobTaskAttributeGroup"/>
</xs:complexType>
```

### 2.6.4.4 ExecutableType Instance for History Cleanup Task

History Cleanup Task deletes history entries from a database. An executable is a History Cleanup Task executable if the **ExecutableType** attribute value is one of the following:

- Microsoft.SqlServer.Management.DatabaseMaintenance.DbMaintenanceHistoryCleanupTask, Microsoft.SqlServer.MaintenancePlanTasks, Version=11.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91 (for DTSX2 2012/01)

- Microsoft.DbMaintenanceHistoryCleanupTask (for DTSX2 2014/01)

The History Cleanup Task executable is formally defined to be of type AnyNonPackageExecutableType. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type. This executable MUST follow the anonymous type declaration that is contained in this section.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **AnyNonPackageExecutableType** type:

- This **Executable** element MUST NOT contain an **Executables** element.

- This **Executable** element MUST NOT contain a **ForEachEnumerator** element.

- This **Executable** element MUST NOT contain a **ForEachVariableMappings** element.

- In the **AnyNonPackageExecutableType** type, the type of the **ObjectData** element is given as ExecutableObjectDataType. The **ExecutableObjectDataType** type definition contains an **xs:choice** XSD Schema element. However, all of the choices that are allowed in the **xs:choice** XSD Schema element are not available for the History Cleanup Task executable.

  For this **Executable** element, the **ObjectData** element MUST contain the SqlTaskData element that is specified in the SQLTask namespace. The **SqlTaskData** element declaration in the **SQLTask** namespace MUST be replaced by the declaration and type that are specified in this section. This instance of the **SqlTaskData** element MUST contain zero elements. This instance of the **SqlTaskData** element MUST NOT contain attributes other than those that are specified in the SqlTaskBaseAttributeGroup attribute group, the MaintenanceTaskBaseAttributeGroup attribute group, the HistoryCleanupTaskAttributeGroup attribute group, the **RemoveOlderThan** attribute, and the **TimeUnitsType** attribute.

- The allowed attributes are a restricted subset of those that are allowed on the type, as specified in the AnyNonPackageExecutableAttributeGroup attribute group. Hence, the declaration of the attributes is replaced in this anonymous XSD fragment. The attributes used MUST be restricted to the ones that are shown as valid in the XSD fragment that is contained in this section.

```
   <xs:complexType>
     <xs:sequence>
       <xs:element name="ForEachEnumerator" type="DTS:ForEachEnumeratorType" minOccurs="0"
maxOccurs="0"/>
       <xs:element name="Variables" type="DTS:VariablesType" minOccurs="0"/>
       <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType" />
       <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
minOccurs="0"/>
       <xs:element name="Executables" type="DTS:ExecutablesType" minOccurs="0"
maxOccurs="0"/>
       <xs:element name="PrecedenceConstraints" type="DTS:PrecedenceConstraintsType"
minOccurs="0"/>
       <xs:element name="ForEachVariableMappings" type="DTS:ForEachVariableMappingsType"
minOccurs="0" maxOccurs="0"/>
       <xs:element name="EventHandlers" type="DTS:EventHandlersType" minOccurs="0"/>
       <xs:element name="ObjectData">
         <xs:complexType>
           <xs:choice>
             <xs:element ref="SQLTask:SqlTaskData"/>
           </xs:choice>
         </xs:complexType>
       </xs:element>
     </xs:sequence>
     <xs:attribute name="ExecutableType" use="required" type="xs:string"/>
     <xs:attribute name="ThreadHint" use="optional" type="xs:int"/>
     <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
     <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
     <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
     <xs:attribute name="ExecutionLocation" type="xs:int" default="0" use="optional"
form="qualified"/>
     <xs:attribute name="ExecutionAddress" type="xs:string" default="" use="optional"
form="qualified"/>
     <xs:attribute name="TaskContact" type="xs:string" default="" use="optional"
form="qualified"/>
   </xs:complexType>
```

The preceding anonymous complex type definition requires that the declaration for the **SqlTaskData** element and its type, in the **SQLTask** namespace, be changed to the following.

```
  <xs:element name="SqlTaskData"
type="SQLTask:SqlTaskDataElementHistoryCleanupTaskType"/>
  <xs:complexType name="SqlTaskDataElementHistoryCleanupTaskType">
    <xs:sequence/>
    <xs:attributeGroup ref="SQLTask:SqlTaskBaseAttributeGroup"/>
    <xs:attributeGroup ref="SQLTask:MaintenanceTaskBaseAttributeGroup"/>
    <xs:attributeGroup ref="SQLTask:HistoryCleanupTaskAttributeGroup"/>
    <xs:attribute name="RemoveOlderThan" type="xs:int"/>
    <xs:attribute name="TimeUnitsType" type="SQLTask:SqlTaskTimeUnitsTypeEnum"/>
  </xs:complexType>
```

### 2.6.4.5 ExecutableType Instance for Maintenance File Cleanup Task

Maintenance File Cleanup Task removes database maintenance files according to user-specified logic. An executable is a Maintenance File Cleanup Task executable if the **ExecutableType** attribute value is one of the following:

- Microsoft.SqlServer.Management.DatabaseMaintenance.DbMaintenanceFileCleanupTask, Microsoft.SqlServer.MaintenancePlanTasks, Version=11.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91 (for DTSX2 2012/01)

- Microsoft.DbMaintenanceFileCleanupTask (for DTSX2 2014/01)

The Maintenance File Cleanup Task executable is formally defined to be of type AnyNonPackageExecutableType. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type. This executable MUST follow the anonymous type declaration that is contained in this section.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **AnyNonPackageExecutableType** type:

- This **Executable** element MUST NOT contain an **Executables** element.

- This **Executable** element MUST NOT contain a **ForEachEnumerator** element.

- This **Executable** element MUST NOT contain a **ForEachVariableMappings** element.

- In the **AnyNonPackageExecutableType** type, the type of the **ObjectData** element is given as ExecutableObjectDataType. The **ExecutableObjectDataType** type definition contains an **xs:choice** XSD Schema element. However, all of the choices that are allowed in the **xs:choice** XSD Schema element are not available for the Maintenance File Cleanup Task executable.

  For this **Executable** element, the **ObjectData** element MUST contain the SqlTaskData element that is specified in the SQLTask namespace. The **SqlTaskData** element declaration in the **SQLTask** namespace MUST be replaced by the declaration and type that is specified in this section. This instance of the **SqlTaskData** element MUST contain zero elements. This instance of the **SqlTaskData** element MUST NOT contain attributes other than those that are specified in the SqlTaskBaseAttributeGroup attribute group, the MaintenanceTaskBaseAttributeGroup attribute group, the MaintenanceFileCleanupTaskAttributeGroup attribute group, the **RemoveOlderThan** attribute, and the **TimeUnitsType** attribute.

- The allowed attributes are a restricted subset of those that are allowed on the type, as specified in the AnyNonPackageExecutableAttributeGroup attribute group. Hence, the declaration of the attributes is replaced in this anonymous XSD fragment. The attributes used MUST be restricted to the ones that are shown as valid in the XSD fragment that is contained in this section.

```
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ForEachEnumerator" type="DTS:ForEachEnumeratorType" minOccurs="0"
maxOccurs="0"/>
```

```
        <xs:element name="Variables" type="DTS:VariablesType" minOccurs="0"/>
        <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType" />
        <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
minOccurs="0"/>
        <xs:element name="Executables" type="DTS:ExecutablesType" minOccurs="0"
maxOccurs="0"/>
        <xs:element name="PrecedenceConstraints" type="DTS:PrecedenceConstraintsType"
minOccurs="0"/>
        <xs:element name="ForEachVariableMappings" type="DTS:ForEachVariableMappingsType"
minOccurs="0" maxOccurs="0"/>
        <xs:element name="EventHandlers" type="DTS:EventHandlersType" minOccurs="0"/>
        <xs:element name="ObjectData">
          <xs:complexType>
            <xs:choice>
              <xs:element ref="SQLTask:SqlTaskData"/>
            </xs:choice>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attribute name="ExecutableType" use="required" type="xs:string"/>
      <xs:attribute name="ThreadHint" use="optional" type="xs:int"/>
      <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
      <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
      <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
      <xs:attribute name="ExecutionLocation" type="xs:int" default="0" use="optional"
form="qualified"/>
      <xs:attribute name="ExecutionAddress" type="xs:string" default="" use="optional"
form="qualified"/>
      <xs:attribute name="TaskContact" type="xs:string" default="" use="optional"
form="qualified"/>
    </xs:complexType>
```

The preceding anonymous complex type definition requires that the declaration for the **SqlTaskData**
element and its type, in the **SQLTask** namespace, be changed to the following.

```
  <xs:element name="SqlTaskData"
type="SQLTask:SqlTaskDataElementMaintenanceFileCleanupTaskType"/>
  <xs:complexType name="SqlTaskDataElementMaintenanceFileCleanupTaskType">
    <xs:sequence/>
    <xs:attributeGroup ref="SQLTask:SqlTaskBaseAttributeGroup"/>
    <xs:attributeGroup ref="SQLTask:MaintenanceTaskBaseAttributeGroup"/>
    <xs:attributeGroup ref="SQLTask:MaintenanceFileCleanupTaskAttributeGroup"/>
    <xs:attribute name="RemoveOlderThan" type="xs:int"/>
    <xs:attribute name="TimeUnitsType" type="SQLTask:SqlTaskTimeUnitsTypeEnum"/>
  </xs:complexType>
```

### 2.6.4.6 ExecutableType Instance for Notify Operator Task

Notify Operator Task sends electronic notification messages to a list of SQL Server operators. An
executable is a Notify Operator Task executable if the **ExecutableType** attribute value is one of the
following:

▪ Microsoft.SqlServer.Management.DatabaseMaintenance.DbMaintenanceNotifyOperatorTask,
  Microsoft.SqlServer.MaintenancePlanTasks, Version=11.0.0.0, Culture=neutral,
  PublicKeyToken=89845dcd8080cc91 (for DTSX2 2012/01)

▪ Microsoft.DbMaintenanceNotifyOperatorTask (for DTSX2 2014/01)

The Notify Operator Task executable is formally defined to be of type AnyNonPackageExecutableType.
However, the following XSD fragment, which is expressed as an anonymous complex type declaration,
places further restrictions on the type. This executable MUST follow the anonymous type declaration
that is contained in this section.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **AnyNonPackageExecutableType** type:

- This **Executable** element MUST NOT contain an **Executables** element.

- This **Executable** element MUST NOT contain a **ForEachEnumerator** element.

- This **Executable** element MUST NOT contain a **ForEachVariableMappings** element.

- In the **AnyNonPackageExecutableType** type, the type of the **ObjectData** element is given as ExecutableObjectDataType. The **ExecutableObjectDataType** type definition contains an **xs:choice** XSD Schema element. However, all of the choices that are allowed in the **xs:choice** XSD Schema element are not available for the Notify Operator Task executable.

  For this **Executable** element, the **ObjectData** element MUST contain the SqlTaskData element that is specified in the SQLTask namespace. The **SqlTaskData** element declaration in the **SQLTask** namespace MUST be replaced by the declaration and type that is specified in this section. This instance of the **SqlTaskData** element MUST NOT contain elements except for the **OperatorNotifyList** element. This instance of the **SqlTaskData** element MUST NOT contain attributes other than those that are specified in the SqlTaskBaseAttributeGroup attribute group, the MaintenanceTaskBaseAttributeGroup attribute group, and the NotifyOperatorTaskAttributeGroup attribute group.

- The allowed attributes are a restricted subset of those that are allowed on the type, as specified in the AnyNonPackageExecutableAttributeGroup attribute group. Hence, the declaration of the attributes is replaced in this anonymous XSD fragment. The attributes used MUST be restricted to the ones that are shown as valid in the XSD fragment that is contained in this section.

```
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ForEachEnumerator" type="DTS:ForEachEnumeratorType" minOccurs="0"
maxOccurs="0"/>
      <xs:element name="Variables" type="DTS:VariablesType" minOccurs="0"/>
      <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType" />
      <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
minOccurs="0"/>
      <xs:element name="Executables" type="DTS:ExecutablesType" minOccurs="0"
maxOccurs="0"/>
      <xs:element name="PrecedenceConstraints" type="DTS:PrecedenceConstraintsType"
minOccurs="0"/>
      <xs:element name="ForEachVariableMappings" type="DTS:ForEachVariableMappingsType"
minOccurs="0" maxOccurs="0"/>
      <xs:element name="EventHandlers" type="DTS:EventHandlersType" minOccurs="0"/>
      <xs:element name="ObjectData">
        <xs:complexType>
          <xs:choice>
            <xs:element ref="SQLTask:SqlTaskData"/>
          </xs:choice>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="ExecutableType" use="required" type="xs:string"/>
    <xs:attribute name="ThreadHint" use="optional" type="xs:int"/>
    <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
    <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
    <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
    <xs:attribute name="ExecutionLocation" type="xs:int" default="0" use="optional"
form="qualified"/>
    <xs:attribute name="ExecutionAddress" type="xs:string" default="" use="optional"
form="qualified"/>
    <xs:attribute name="TaskContact" type="xs:string" default="" use="optional"
form="qualified"/>
  </xs:complexType>
```

The preceding anonymous complex type definition requires that the declaration for the **SqlTaskData** element and its type, in the **SQLTask** namespace, be changed to the following.

```
   <xs:element name="SqlTaskData"
type="SQLTask:SqlTaskDataElementNotifyOperatorTaskType"/>
   <xs:complexType name="SqlTaskDataElementNotifyOperatorTaskType">
     <xs:sequence>
       <xs:element name="OperatorNotifyList" type="SQLTask:OperatorNotifyListType"
minOccurs="0" maxOccurs="unbounded"/>
     </xs:sequence>
     <xs:attributeGroup ref="SQLTask:SqlTaskBaseAttributeGroup"/>
     <xs:attributeGroup ref="SQLTask:MaintenanceTaskBaseAttributeGroup"/>
     <xs:attributeGroup ref="SQLTask:NotifyOperatorTaskAttributeGroup"/>
   </xs:complexType>
```

### 2.6.4.7 ExecutableType Instance for Rebuild Index Task

Rebuild Index Task rebuilds indexes in database tables and views. An executable is a Rebuild Index Task executable if the **ExecutableType** attribute value is one of the following:

- Microsoft.SqlServer.Management.DatabaseMaintenance.DbMaintenanceReindexTask, Microsoft.SqlServer.MaintenancePlanTasks, Version=11.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91 (for DTSX2 2012/01)

- Microsoft.DbMaintenanceReindexTask (for DTSX2 2014/01)

The Rebuild Index Task executable is formally defined to be of type AnyNonPackageExecutableType. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type. This executable MUST follow the anonymous type declaration that is contained in this section.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **AnyNonPackageExecutableType** type:

- This **Executable** element MUST NOT contain an **Executables** element.

- This **Executable** element MUST NOT contain a **ForEachEnumerator** element.

- This **Executable** element MUST NOT contain a **ForEachVariableMappings** element.

- In the **AnyNonPackageExecutableType** type, the type of the **ObjectData** element is given as ExecutableObjectDataType. The **ExecutableObjectDataType** type definition contains an **xs:choice** XSD Schema element. However, all of the choices that are allowed in the **xs:choice** XSD Schema element are not available for the Rebuild Index Task executable.

  For this **Executable** element, the **ObjectData** element MUST contain the SqlTaskData element that is specified in the SQLTask namespace. The **SqlTaskData** element declaration in the **SQLTask** namespace MUST be replaced by the declaration and type that is specified in this section. This instance of the **SqlTaskData** element MUST NOT contain elements except the **SelectedDatabases** element. This instance of the **SqlTaskData** element MUST NOT contain attributes other than those that are specified in the SqlTaskBaseAttributeGroup attribute group, the MaintenanceTaskBaseAttributeGroup attribute group, the RebuildIndexTaskAttributeGroup attribute group, and the **DatabaseSelectionType** attribute.

- The allowed attributes are a restricted subset of those that are allowed on the type, as specified in the AnyNonPackageExecutableAttributeGroup attribute group. Hence, the declaration of the attributes is replaced in this anonymous XSD fragment. The attributes used MUST be restricted to the ones that are shown as valid in the XSD fragment that is contained in this section.

```
   <xs:complexType>
```

```
    <xs:sequence>
      <xs:element name="ForEachEnumerator" type="DTS:ForEachEnumeratorType" minOccurs="0"
maxOccurs="0"/>
      <xs:element name="Variables" type="DTS:VariablesType" minOccurs="0"/>
      <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType" />
      <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
minOccurs="0"/>
      <xs:element name="Executables" type="DTS:ExecutablesType" minOccurs="0"
maxOccurs="0"/>
      <xs:element name="PrecedenceConstraints" type="DTS:PrecedenceConstraintsType"
minOccurs="0"/>
      <xs:element name="ForEachVariableMappings" type="DTS:ForEachVariableMappingsType"
minOccurs="0" maxOccurs="0"/>
      <xs:element name="EventHandlers" type="DTS:EventHandlersType" minOccurs="0"/>
      <xs:element name="ObjectData">
        <xs:complexType>
          <xs:choice>
            <xs:element ref="SQLTask:SqlTaskData"/>
          </xs:choice>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="ExecutableType" use="required" type="xs:string"/>
    <xs:attribute name="ThreadHint" use="optional" type="xs:int"/>
    <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
    <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
    <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
    <xs:attribute name="ExecutionLocation" type="xs:int" default="0" use="optional"
form="qualified"/>
    <xs:attribute name="ExecutionAddress" type="xs:string" default="" use="optional"
form="qualified"/>
    <xs:attribute name="TaskContact" type="xs:string" default="" use="optional"
form="qualified"/>
  </xs:complexType>
```

The preceding anonymous complex type definition requires that the declaration for the **SqlTaskData** element and its type, in the **SQLTask** namespace, be changed to the following.

```
    <xs:element name="SqlTaskData" type="SQLTask:SqlTaskDataElementRebuildIndexTaskType"/>
    <xs:complexType name="SqlTaskDataElementRebuildIndexTaskType">
      <xs:sequence>
        <xs:element name="SelectedDatabases" type="SQLTask:SelectedDatabasesType"
minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attributeGroup ref="SQLTask:SqlTaskBaseAttributeGroup"/>
      <xs:attributeGroup ref="SQLTask:MaintenanceTaskBaseAttributeGroup"/>
      <xs:attributeGroup ref="SQLTask:RebuildIndexTaskAttributeGroup"/>
      <xs:attribute name="DatabaseSelectionType"
type="SQLTask:SqlTaskDatabaseSelectionTypeEnum"/>
    </xs:complexType>
```

## 2.6.4.8 ExecutableType Instance for Reorganize Index Task

Reorganize Index Task defragments an index in a database table or view. An executable is a Reorganize Index Task executable if the **ExecutableType** attribute value is one of the following:

- Microsoft.SqlServer.Management.DatabaseMaintenance.DbMaintenanceDefragmentIndexTask, Microsoft.SqlServer.MaintenancePlanTasks, Version=11.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91 (for DTSX2 2012/01)

- Microsoft.DbMaintenanceDefragmentIndexTask (for DTSX2 2014/01)

The Reorganize Index Task executable is formally defined to be of type AnyNonPackageExecutableType. However, the following XSD fragment, which is expressed as an

anonymous complex type declaration, places further restrictions on the type. This executable MUST follow the anonymous type declaration that is contained in this section.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **AnyNonPackageExecutableType** type:

- This **Executable** element MUST NOT contain an **Executables** element.

- This **Executable** element MUST NOT contain a **ForEachEnumerator** element.

- This **Executable** element MUST NOT contain a **ForEachVariableMappings** element.

- In the **AnyNonPackageExecutableType** type, the type of the **ObjectData** element is given as ExecutableObjectDataType. The **ExecutableObjectDataType** type definition contains an **xs:choice** XSD Schema element. However, all of the choices that are allowed in the **xs:choice** XSD Schema element are not available for the Reorganize Index Task executable.

  For this **Executable** element, the **ObjectData** element MUST contain the SqlTaskData element that is specified in the SQLTask namespace. The **SqlTaskData** element declaration in the **SQLTask** namespace MUST be replaced by the declaration and type that is specified in this section. This instance of the **SqlTaskData** element MUST NOT contain elements except for the **SelectedDatabases** element. This instance of the **SqlTaskData** element MUST NOT contain attributes other than those that are in the SqlTaskBaseAttributeGroup attribute group, the MaintenanceTaskBaseAttributeGroup attribute group, the **ReorganizeIndexTaskAttributeGroup** attribute group, and the **DatabaseSelectionType** attribute.

- The allowed attributes are a restricted subset of those that are allowed on the type, as specified in the AnyNonPackageExecutableAttributeGroup attribute group. Hence, the declaration of the attributes is replaced in this anonymous XSD fragment. The attributes used MUST be restricted to the ones that are shown as valid in the XSD fragment that is contained in this section.

```
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ForEachEnumerator" type="DTS:ForEachEnumeratorType" minOccurs="0"
maxOccurs="0"/>
      <xs:element name="Variables" type="DTS:VariablesType" minOccurs="0"/>
      <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType" />
      <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
minOccurs="0"/>
      <xs:element name="Executables" type="DTS:ExecutablesType" minOccurs="0"
maxOccurs="0"/>
      <xs:element name="PrecedenceConstraints" type="DTS:PrecedenceConstraintsType"
minOccurs="0"/>
      <xs:element name="ForEachVariableMappings" type="DTS:ForEachVariableMappingsType"
minOccurs="0" maxOccurs="0"/>
      <xs:element name="EventHandlers" type="DTS:EventHandlersType" minOccurs="0"/>
      <xs:element name="ObjectData">
        <xs:complexType>
          <xs:choice>
            <xs:element ref="SQLTask:SqlTaskData"/>
          </xs:choice>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="ExecutableType" use="required" type="xs:string"/>
    <xs:attribute name="ThreadHint" use="optional" type="xs:int"/>
    <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
    <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
    <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
    <xs:attribute name="ExecutionLocation" type="xs:int" default="0" use="optional"
form="qualified"/>
    <xs:attribute name="ExecutionAddress" type="xs:string" default="" use="optional"
form="qualified"/>
```

```
      <xs:attribute name="TaskContact" type="xs:string" default="" use="optional"
  form="qualified"/>
      </xs:complexType>
```

The preceding anonymous complex type definition requires that the declaration for the **SqlTaskData** element and its type, in the **SQLTask** namespace, be changed to the following.

```
    <xs:element name="SqlTaskData"
  type="SQLTask:SqlTaskDataElementReorganizeIndexTaskType"/>
      <xs:complexType name="SqlTaskDataElementReorganizeIndexTaskType">
        <xs:sequence>
          <xs:element name="SelectedDatabases" type="SQLTask:SelectedDatabasesType"
  minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attributeGroup ref="SQLTask:SqlTaskBaseAttributeGroup"/>
        <xs:attributeGroup ref="SQLTask:MaintenanceTaskBaseAttributeGroup"/>
        <xs:attributeGroup ref="SQLTask:ReorganizeIndexTaskAttributeGroup"/>
        <xs:attribute name="DatabaseSelectionType"
  type="SQLTask:SqlTaskDatabaseSelectionTypeEnum"/>
      </xs:complexType>
```

### 2.6.4.9 ExecutableType Instance for Shrink Database Task

Shrink Database Task removes unused space from within database files. An executable is a Shrink Database Task executable if the **ExecutableType** attribute value is one of the following:

▪ Microsoft.SqlServer.Management.DatabaseMaintenance.DbMaintenanceShrinkTask, Microsoft.SqlServer.MaintenancePlanTasks, Version=11.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91 (for DTSX2 2012/01)

▪ Microsoft.DbMaintenanceShrinkTask (for DTSX2 2014/01)

The Shrink Database Task executable is formally defined to be of type AnyNonPackageExecutableType. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type. This executable MUST follow the anonymous type declaration that is contained in this section.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **AnyNonPackageExecutableType** type:

▪ This **Executable** element MUST NOT contain an **Executables** element.

▪ This **Executable** element MUST NOT contain a **ForEachEnumerator** element.

▪ This **Executable** element MUST NOT contain a **ForEachVariableMappings** element.

▪ In the **AnyNonPackageExecutableType** type, the type of the **ObjectData** element is given as ExecutableObjectDataType. The **ExecutableObjectDataType** type definition contains an **xs:choice** XSD Schema element. However, all of the choices that are allowed in the **xs:choice** XSD Schema element are not available for the Shrink Database Task executable.

For this **Executable** element, the **ObjectData** element MUST contain the SqlTaskData element that is specified in the SQLTask namespace. The **SqlTaskData** element declaration in the **SQLTask** namespace MUST be replaced by the declaration and type that is specified in this section. This instance of the **SqlTaskData** element MUST NOT contain elements except for the **SelectedDatabases** element. This instance of the **SqlTaskData** element MUST NOT contain attributes other than those that are specified in the SqlTaskBaseAttributeGroup attribute group, the MaintenanceTaskBaseAttributeGroup attribute group, the ShrinkDBTaskAttributeGroup attribute group, and the **DatabaseSelectionType** attribute.

- The allowed attributes are a restricted subset of those that are allowed on the type, as specified in the AnyNonPackageExecutableAttributeGroup attribute group. Hence, the declaration of the attributes is replaced in this anonymous XSD fragment. The attributes used MUST be restricted to the ones that are shown as valid in the XSD fragment that is contained in this section.

```
<xs:complexType>
  <xs:sequence>
    <xs:element name="ForEachEnumerator" type="DTS:ForEachEnumeratorType" minOccurs="0"
maxOccurs="0"/>
    <xs:element name="Variables" type="DTS:VariablesType" minOccurs="0"/>
    <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType" />
    <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
minOccurs="0"/>
    <xs:element name="Executables" type="DTS:ExecutablesType" minOccurs="0"
maxOccurs="0"/>
    <xs:element name="PrecedenceConstraints" type="DTS:PrecedenceConstraintsType"
minOccurs="0"/>
    <xs:element name="ForEachVariableMappings" type="DTS:ForEachVariableMappingsType"
minOccurs="0" maxOccurs="0"/>
    <xs:element name="EventHandlers" type="DTS:EventHandlersType" minOccurs="0"/>
    <xs:element name="ObjectData">
      <xs:complexType>
        <xs:choice>
          <xs:element ref="SQLTask:SqlTaskData"/>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="ExecutableType" use="required" type="xs:string"/>
  <xs:attribute name="ThreadHint" use="optional" type="xs:int"/>
  <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
  <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
  <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
  <xs:attribute name="ExecutionLocation" type="xs:int" default="0" use="optional"
form="qualified"/>
  <xs:attribute name="ExecutionAddress" type="xs:string" default="" use="optional"
form="qualified"/>
  <xs:attribute name="TaskContact" type="xs:string" default="" use="optional"
form="qualified"/>
</xs:complexType>
```

The preceding anonymous complex type definition requires that the declaration for the **SqlTaskData** element and its type, in the **SQLTask** namespace, be changed to the following.

```
<xs:element name="SqlTaskData"
type="SQLTask:SqlTaskDataElementShrinkDatabaseTaskType"/>
  <xs:complexType name="SqlTaskDataElementShrinkDatabaseTaskType">
    <xs:sequence>
      <xs:element name="SelectedDatabases" type="SQLTask:SelectedDatabasesType"
minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attributeGroup ref="SQLTask:SqlTaskBaseAttributeGroup"/>
    <xs:attributeGroup ref="SQLTask:MaintenanceTaskBaseAttributeGroup"/>
    <xs:attributeGroup ref="SQLTask:ShrinkDBTaskAttributeGroup"/>
    <xs:attribute name="DatabaseSelectionType"
type="SQLTask:SqlTaskDatabaseSelectionTypeEnum"/>
  </xs:complexType>
```

### 2.6.4.10    ExecutableType Instance for Execute TSQL Task

Execute TSQL Task runs Transact-SQL (TSQL) language statements from a package. An executable is an Execute TSQL Task executable if the **ExecutableType** attribute value is one of the following:

- Microsoft.SqlServer.Management.DatabaseMaintenance.DbMaintenanceTSQLExecuteTask, Microsoft.SqlServer.MaintenancePlanTasks, Version=11.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91 (for DTSX2 2012/01)

- Microsoft.DbMaintenanceTSQLExecuteTask (for DTSX2 2014/01)

The Execute TSQL Task executable is formally defined to be of type AnyNonPackageExecutableType. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type. This executable MUST follow the anonymous type declaration that is contained in this section.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **AnyNonPackageExecutableType** type:

- This **Executable** element MUST NOT contain an **Executables** element.

- This **Executable** element MUST NOT contain a **ForEachEnumerator** element.

- This **Executable** element MUST NOT contain a **ForEachVariableMappings** element.

- In the **AnyNonPackageExecutableType** type, the type of the **ObjectData** element is given as ExecutableObjectDataType. The **ExecutableObjectDataType** type definition contains an **xs:choice** XSD Schema element. However, all of the choices that are allowed in the **xs:choice** XSD Schema element are not available for the Execute TSQL Task executable.

  For this **Executable** element, the **ObjectData** element MUST contain the SqlTaskData element that is specified in the SQLTask namespace. The **SqlTaskData** element declaration in the **SQLTask** namespace MUST be replaced by the declaration and type that is specified in this section. This instance of the **SqlTaskData** element MUST contain zero elements. This instance of the **SqlTaskData** element MUST NOT contain attributes other than those that are specified in the SqlTaskBaseAttributeGroup attribute group and the MaintenanceTaskBaseAttributeGroup attribute group.

- The allowed attributes are a restricted subset of those that are allowed on the type, as specified in the AnyNonPackageExecutableAttributeGroup attribute group. Hence, the declaration of the attributes is replaced in this anonymous XSD fragment. The attributes used MUST be restricted to the ones that are shown as valid in the XSD fragment that is contained in this section.

```
   <xs:complexType>
     <xs:sequence>
       <xs:element name="ForEachEnumerator" type="DTS:ForEachEnumeratorType" minOccurs="0"
 maxOccurs="0"/>
       <xs:element name="Variables" type="DTS:VariablesType" minOccurs="0"/>
       <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType" />
       <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
 minOccurs="0"/>
       <xs:element name="Executables" type="DTS:ExecutablesType" minOccurs="0"
 maxOccurs="0"/>
       <xs:element name="PrecedenceConstraints" type="DTS:PrecedenceConstraintsType"
 minOccurs="0"/>
       <xs:element name="ForEachVariableMappings" type="DTS:ForEachVariableMappingsType"
 minOccurs="0" maxOccurs="0"/>
       <xs:element name="EventHandlers" type="DTS:EventHandlersType" minOccurs="0"/>
       <xs:element name="ObjectData">
         <xs:complexType>
           <xs:choice>
             <xs:element ref="SQLTask:SqlTaskData"/>
           </xs:choice>
         </xs:complexType>
       </xs:element>
     </xs:sequence>
     <xs:attribute name="ExecutableType" use="required" type="xs:string"/>
     <xs:attribute name="ThreadHint" use="optional" type="xs:int"/>
```

```
    <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
    <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
    <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
    <xs:attribute name="ExecutionLocation" type="xs:int" default="0" use="optional"
form="qualified"/>
    <xs:attribute name="ExecutionAddress" type="xs:string" default="" use="optional"
form="qualified"/>
    <xs:attribute name="TaskContact" type="xs:string" default="" use="optional"
form="qualified"/>
 </xs:complexType>
```

The preceding anonymous complex type definition requires that the declaration for the **SqlTaskData** element and its type, in the **SQLTask** namespace, be changed to the following.

```
    <xs:element name="SqlTaskData" type="SQLTask:SqlTaskDataElementTSQLExecuteTaskType"/>
    <xs:complexType name="SqlTaskDataElementTSQLExecuteTaskType">
      <xs:sequence/>
      <xs:attributeGroup ref="SQLTask:SqlTaskBaseAttributeGroup"/>
      <xs:attributeGroup ref="SQLTask:MaintenanceTaskBaseAttributeGroup"/>
    </xs:complexType>
```

### 2.6.4.11    ExecutableType Instance for Update Statistics Task

Update Statistics Task updates the statistics about key columns that are stored in a database. An executable is an Update Statistics Task executable if the **ExecutableType** attribute value is one of the following:

▪ Microsoft.SqlServer.Management.DatabaseMaintenance.DbMaintenanceUpdateStatisticsTask, Microsoft.SqlServer.MaintenancePlanTasks, Version=11.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91 (for DTSX2 2012/01)

▪ Microsoft.DbMaintenanceUpdateStatisticsTask (for DTSX2 2014/01)

The Update Statistics Task executable is formally defined to be of type AnyNonPackageExecutableType. However, the following XSD fragment, which is expressed as an anonymous complex type declaration, places further restrictions on the type. This executable MUST follow the anonymous type declaration that is contained in this section.

Note the following differences between this anonymous complex type declaration and the full definition of the complex type for the **AnyNonPackageExecutableType** type:

▪ This **Executable** element MUST NOT contain an **Executables** element.

▪ This **Executable** element MUST NOT contain a **ForEachEnumerator** element.

▪ This **Executable** element MUST NOT contain a **ForEachVariableMappings** element.

▪ In the **AnyNonPackageExecutableType** type, the type of the **ObjectData** element is given as ExecutableObjectDataType. The **ExecutableObjectDataType** type definition contains an **xs:choice** XSD Schema element. However, all of the choices that are allowed in the **xs:choice** XSD Schema element are not available for the Update Statistics Task executable.

For this **Executable** element, the **ObjectData** element MUST contain the SqlTaskData element that is specified in the SQLTask namespace. The **SqlTaskData** element declaration in the **SQLTask** namespace MUST be replaced by the declaration and type that is specified in this section. This instance of the **SqlTaskData** element MUST NOT contain elements except for the **SelectedDatabases** element. This instance of the **SqlTaskData** element MUST NOT contain attributes other than those that are specified in the SqlTaskBaseAttributeGroup attribute group, the MaintenanceTaskBaseAttributeGroup attribute group, the UpdateStatisticsTaskAttributeGroup attribute group, and the **DatabaseSelectionType** attribute.

- The allowed attributes are a restricted subset of those that are allowed on the type, as specified in the AnyNonPackageExecutableAttributeGroup attribute group. Hence, the declaration of the attributes is replaced in this anonymous XSD fragment. The attributes used MUST be restricted to the ones that are shown as valid in the XSD fragment that is contained in this section.

```
<xs:complexType>
  <xs:sequence>
    <xs:element name="ForEachEnumerator" type="DTS:ForEachEnumeratorType" minOccurs="0"
maxOccurs="0"/>
    <xs:element name="Variables" type="DTS:VariablesType" minOccurs="0"/>
    <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType" />
    <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
minOccurs="0"/>
    <xs:element name="Executables" type="DTS:ExecutablesType" minOccurs="0"
maxOccurs="0"/>
    <xs:element name="PrecedenceConstraints" type="DTS:PrecedenceConstraintsType"
minOccurs="0"/>
    <xs:element name="ForEachVariableMappings" type="DTS:ForEachVariableMappingsType"
minOccurs="0" maxOccurs="0"/>
    <xs:element name="EventHandlers" type="DTS:EventHandlersType" minOccurs="0"/>
    <xs:element name="ObjectData">
      <xs:complexType>
        <xs:choice>
          <xs:element ref="SQLTask:SqlTaskData"/>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="ExecutableType" use="required" type="xs:string"/>
  <xs:attribute name="ThreadHint" use="optional" type="xs:int"/>
  <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
  <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
  <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
  <xs:attribute name="ExecutionLocation" type="xs:int" default="0" use="optional"
form="qualified"/>
  <xs:attribute name="ExecutionAddress" type="xs:string" default="" use="optional"
form="qualified"/>
  <xs:attribute name="TaskContact" type="xs:string" default="" use="optional"
form="qualified"/>
</xs:complexType>
```

The preceding anonymous complex type definition requires that the declaration for the **SqlTaskData** element and its type, in the **SQLTask** namespace, be changed to the following.

```
<xs:element name="SqlTaskData"
type="SQLTask:SqlTaskDataElementUpdateStatisticsTaskType"/>
  <xs:complexType
    name="SqlTaskDataElementUpdateStatisticsTaskType">
    <xs:sequence>
      <xs:element name="SelectedDatabases" type="SQLTask:SelectedDatabasesType"
minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attributeGroup ref="SQLTask:SqlTaskBaseAttributeGroup"/>
    <xs:attributeGroup ref="SQLTask:MaintenanceTaskBaseAttributeGroup"/>
    <xs:attributeGroup ref="SQLTask:UpdateStatisticsTaskAttributeGroup"/>
    <xs:attribute name="DatabaseSelectionType"
type="SQLTask:SqlTaskDatabaseSelectionTypeEnum"/>
  </xs:complexType>
```

## 2.7 Executable ObjectData Types

### 2.7.1 ExecutableObjectDataType

The **ExecutableObjectDataType** type is the type of the **ObjectData** element that occurs as a child element of many Executable elements. The choice made within the **xs:Choice** XSD element is constrained by the value of the **ExecutableType** attribute on the **Executable** element. These constraints MUST be adhered to. The constraint for each **ExecutableType** value is explained in ExecutableType Attribute Values for Different Executables.

The following is the XSD of the **ExecutableObjectDataType** type.

*Applies to DTSX2 schema for version 2012/01 (DTSX2 2012/01)*

```
<xs:complexType name="ExecutableObjectDataType">
  <xs:choice minOccurs="0">
    <xs:element name="pipeline" type="DTS:pipelineObjectDataType"
                form="unqualified"/>
    <xs:element name="CDCControlTask" form="unqualified"
                type="DTS:CDCControlTaskObjectDataType"/>
    <xs:element ref="ActiveXScriptTask:ActiveXScriptTaskData" />
    <xs:element name="ASDdlData" type="DTS:ASDdlDataObjectDataType"
                form="unqualified"/>
    <xs:element name="ASProcessingData" form="unqualified"
                type="DTS:ASProcessingDataObjectDataType"/>
    <xs:element ref="BulkInsertTask:BulkInsertTaskData"/>
    <xs:element name="DMQueryTaskData"  form="unqualified"
                type="DTS:DMQueryTaskDataObjectDataType"/>
    <xs:element name="DataProfilingTaskData"  form="unqualified"
                type="DTS:DataProfilingTaskDataObjectDataType"/>
    <xs:element ref="Exec80PackageTask:Exec80PackageTaskData"/>
    <xs:element name="ExecutePackageTask"  form="unqualified"
                type="DTS:ExecutePackageTaskObjectDataType"/>
    <xs:element name="ExecuteProcessData" form="unqualified"
                type="DTS:ExecuteProcessDataObjectDataType"/>
    <xs:element name="ExpressionTask" form="unqualified"
                type="DTS:ExpressionTaskObjectDataType"/>
    <xs:element ref="SQLTask:SqlTaskData"/>
    <xs:element name="FileSystemData" form="unqualified"
                type="DTS:FileSystemDataObjectDataType"/>
    <xs:element name="FtpData" type="DTS:FtpDataObjectDataType"
                form="unqualified"/>
    <xs:element ref="MessageQueueTask:MessageQueueTaskData"/>
    <xs:element name="ScriptProject" type="DTS:ScriptProjectObjectDataType"
                form="unqualified"/>
    <xs:element ref="SendMailTask:SendMailTaskData"/>
    <xs:element name="TransferDatabasesTaskData" form="unqualified"
                type="DTS:TransferDatabasesTaskDataObjectDataType"/>
    <xs:element name="TransferErrorMessagesTaskData" form="unqualified"
                type="DTS:TransferErrorMessagesTaskDataObjectDataType"/>
    <xs:element name="TransferJobsTaskData" form="unqualified"
                type="DTS:TransferJobsTaskDataObjectDataType"/>
    <xs:element name="TransferLoginsTaskData" form="unqualified"
                type="DTS:TransferLoginsTaskDataObjectDataType"/>
    <xs:element name="TransferStoredProceduresTaskData" form="unqualified"
                type="DTS:TransferStoredProceduresTaskDataObjectDataType"/>
    <xs:element name="TransferSqlServerObjectsTaskData" form="unqualified"
                type="DTS:TransferSqlServerObjectsTaskDataObjectDataType"/>
    <xs:element ref="WSTask:WebServiceTaskData"/>
    <xs:element name="WMIDRTaskData" type="DTS:WMIDRTaskDataObjectDataType"
                form="unqualified"/>
    <xs:element name="WMIEWTaskData" type="DTS:WMIEWTaskDataObjectDataType"
                form="unqualified"/>
    <xs:element name="XMLTaskData" type="DTS:XMLTaskDataObjectDataType"
                form="unqualified"/>
    <xs:element name="InnerObject" type="DTS:InnerObjectObjectDataType"
```

```
                        form="unqualified"/>
        </xs:choice>
    </xs:complexType>
```

*Applies to DTSX2 schema for version 2014/01 (DTSX2 2014/01)*

```
        <xs:complexType name="ExecutableObjectDataType">
          <xs:choice minOccurs="0">
            <xs:element name="pipeline" type="DTS:pipelineObjectDataType"
                        form="unqualified"/>
            <xs:element name="CDCControlTask" form="unqualified"
                        type="DTS:CDCControlTaskObjectDataType"/>
            <xs:element name="ASDdlData" type="DTS:ASDdlDataObjectDataType"
                        form="unqualified"/>
            <xs:element name="ASProcessingData" form="unqualified"
                        type="DTS:ASProcessingDataObjectDataType"/>
            <xs:element ref="BulkInsertTask:BulkInsertTaskData"/>
            <xs:element name="DMQueryTaskData"  form="unqualified"
                        type="DTS:DMQueryTaskDataObjectDataType"/>
            <xs:element name="DataProfilingTaskData"  form="unqualified"
                        type="DTS:DataProfilingTaskDataObjectDataType"/>
            <xs:element name="ExecutePackageTask"  form="unqualified"
                        type="DTS:ExecutePackageTaskObjectDataType"/>
            <xs:element name="ExecuteProcessData" form="unqualified"
                        type="DTS:ExecuteProcessDataObjectDataType"/>
            <xs:element name="ExpressionTask" form="unqualified"
                        type="DTS:ExpressionTaskObjectDataType"/>
            <xs:element ref="SQLTask:SqlTaskData"/>
            <xs:element name="FileSystemData" form="unqualified"
                        type="DTS:FileSystemDataObjectDataType"/>
            <xs:element name="FtpData" type="DTS:FtpDataObjectDataType"
                        form="unqualified"/>
            <xs:element ref="MessageQueueTask:MessageQueueTaskData"/>
            <xs:element name="ScriptProject" type="DTS:ScriptProjectObjectDataType"
                        form="unqualified"/>
            <xs:element ref="SendMailTask:SendMailTaskData"/>
            <xs:element name="TransferDatabasesTaskData" form="unqualified"
                        type="DTS:TransferDatabasesTaskDataObjectDataType"/>
            <xs:element name="TransferErrorMessagesTaskData" form="unqualified"
                        type="DTS:TransferErrorMessagesTaskDataObjectDataType"/>
            <xs:element name="TransferJobsTaskData" form="unqualified"
                        type="DTS:TransferJobsTaskDataObjectDataType"/>
            <xs:element name="TransferLoginsTaskData" form="unqualified"
                        type="DTS:TransferLoginsTaskDataObjectDataType"/>
            <xs:element name="TransferStoredProceduresTaskData" form="unqualified"
                        type="DTS:TransferStoredProceduresTaskDataObjectDataType"/>
            <xs:element name="TransferSqlServerObjectsTaskData" form="unqualified"
                        type="DTS:TransferSqlServerObjectsTaskDataObjectDataType"/>
            <xs:element ref="WSTask:WebServiceTaskData"/>
            <xs:element name="WMIDRTaskData" type="DTS:WMIDRTaskDataObjectDataType"
                        form="unqualified"/>
            <xs:element name="WMIEWTaskData" type="DTS:WMIEWTaskDataObjectDataType"
                        form="unqualified"/>
            <xs:element name="XMLTaskData" type="DTS:XMLTaskDataObjectDataType"
                        form="unqualified"/>
            <xs:element name="InnerObject" type="DTS:InnerObjectObjectDataType"
                        form="unqualified"/>
          </xs:choice>
        </xs:complexType>
```

### 2.7.1.1  pipelineObjectDataType

The **pipelineObjectDataType** complex type is used to specify the parameters, properties, and settings that are specific to a Pipeline Task executable.

The following is the XSD for the **pipelineObjectDataType** complex type.

```
    <xs:complexType name="pipelineObjectDataType">
      <xs:sequence>
        <xs:sequence>
          <xs:element name="components" type="DTS:PipelineComponentsType"
form="unqualified"/>
          <xs:element name="paths" type="DTS:PipelinePathsType" minOccurs="0"
form="unqualified"/>
        </xs:sequence>
      </xs:sequence>
      <xs:attribute name="id" type="xs:string" form="unqualified" use="required"/>
      <xs:attribute name="name" type="xs:string" form="unqualified" use="optional"
default=""/>
      <xs:attribute name="version" type="xs:int" form="unqualified" use="optional"
fixed="1"/>
      <xs:attribute name="description" type="xs:string" use="optional" default=""
form="unqualified"/>
      <xs:attribute name="defaultBufferMaxRows" type="xs:int" use="optional"
default="10000" form="unqualified"/>
      <xs:attribute name="engineThreads" type="xs:int" use="optional" default="10"
form="unqualified"/>
      <xs:attribute name="defaultBufferSize" type="xs:int" use="optional"
default="10485760" form="unqualified"/>
      <xs:attribute name="BLOBTempStoragePath" type="xs:string" use="optional" default=""
form="unqualified"/>
      <xs:attribute name="bufferTempStoragePath" type="xs:string" use="optional" default=""
form="unqualified"/>
      <xs:attribute name="runInOptimizedMode" type="xs:boolean" use="optional"
default="true" form="unqualified"/>
    </xs:complexType>
```

The following table specifies the elements of the **pipelineObjectDataType** complex type.

| Element | Type definition | Description |
|---|---|---|
| components | PipelineComponentsType | A complex type that specifies a collection of **component** elements. Each **component** element represents one node of the pipeline process. |
| paths | PipelinePathsType | A complex type that specifies a collection of **path** elements. Each **path** element represents one path of data flow within the pipeline process. |

The following table specifies the attributes of the **pipelineObjectDataType** complex type.

| Attribute | Description |
|---|---|
| id | A string value that specifies the identifier for this **pipeline** element. Other elements can refer to this **pipeline** element by using this **id** value. |
| name | A string value that specifies the name for this **pipeline** element. |
| version | An integer value that specifies the version. A value of 1 is expected. |
| description | A string value that specifies the description for this **pipeline** element. |
| defaultBufferMaxRows | An integer value that specifies the maximum number of rows for the default buffer.[26] |
| engineThreads | An integer value that specifies a suggestion to the data flow engine about the number of threads to use.[27] |
| defaultBufferSize | An integer value that specifies the number of bytes to allocate by default for temporary buffer storage.[28] |

| Attribute | Description |
|---|---|
| BLOBTempStoragePath | A string value that specifies a folder that is used for temporary storage of BLOB columns during the execution of the task.<29> |
| bufferTempStoragePath | A string value that specifies a folder that is used for temporary buffer storage during the execution of the task.<30> |
| runInOptimizedMode | A Boolean value that specifies whether to run the task in optimized mode. Optimized mode removes unused columns, outputs, and components from the data flow.<br><br>TRUE specifies that the task is run in optimized mode.<br><br>FALSE specifies that the task is not run in optimized mode. |

### 2.7.1.1.1 PipelineComponentsType

The **PipelineComponentsType** complex type specifies a collection of component objects for a Pipeline Task executable.

The following is the XSD for the **PipelineComponentsType** complex type.

```
    <xs:complexType name="PipelineComponentsType">
      <xs:sequence>
        <xs:element name="component" type="DTS:PipelineComponentType" form="unqualified"
maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
```

The following table specifies the elements of the **PipelineComponentsType** complex type.

| Element | Type definition | Description |
|---|---|---|
| Component | PipelineComponentType | A complex type that specifies a single component within a pipeline. |

### 2.7.1.1.1.1 PipelineComponentType

The **PipelineComponentType** complex type specifies a single component within a Pipeline Task executable. Each component represents a single operation within a Pipeline Task.

The following is the XSD for the **PipelineComponentType** complex type.

```
    <xs:complexType name="PipelineComponentType">
      <xs:sequence>
        <xs:element name="properties" form="unqualified"
type="DTS:PipelineComponentPropertiesType" minOccurs="0"/>
        <xs:element name="connections" form="unqualified"
type="DTS:PipelineComponentConnectionsType" minOccurs="0"/>
        <xs:element name="inputs" form="unqualified" type="DTS:PipelineComponentInputsType"
minOccurs="0"/>
        <xs:element name="outputs" form="unqualified"
type="DTS:PipelineComponentOutputsType" minOccurs="0"/>
      </xs:sequence>
      <xs:attribute name="name" type="xs:string" form="unqualified" use="optional"
default=""/>
      <xs:attribute name="refId" type="xs:string" use="required" form="unqualified"/>
      <xs:attribute name="componentClassID" use="required"
type="DTS:PipelineComponentComponentClassIDEnum" form="unqualified"/>
```

```
        <xs:attribute name="description" type="xs:string" form="unqualified" use="optional"
default=""/>
        <xs:attribute name="localeId" type="xs:int" use="optional" default="-1"
form="unqualified"/>
        <xs:attribute name="usesDispositions" type="xs:boolean" use="optional"
default="false" form="unqualified"/>
        <xs:attribute name="validateExternalMetadata" form="unqualified" use="optional"
default="True">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="True"/>
              <xs:enumeration value="False"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="version" type="xs:int" use="optional" default="0"
form="unqualified"/>
        <xs:attribute name="pipelineVersion" type="xs:int" use="optional" default="0"
form="unqualified"/>
        <xs:attribute name="contactInfo" type="xs:string" use="optional" default=""
form="unqualified"/>
  </xs:complexType>
```

The following table specifies the elements of the **PipelineComponentType** complex type.

| Element | Type definition | Description |
|---------|-----------------|-------------|
| properties | PipelineComponentPropertiesType | A complex type that specifies a collection of **property** elements for the component. |
| connections | PipelineComponentConnectionsType | A complex type that specifies a collection of **connection** elements for the component. |
| inputs | PipelineComponentInputsType | A complex type that specifies a collection of **input** elements that represent inputs arriving at this component. |
| outputs | PipelineComponentOutputsType | A complex type that specifies a collection of **output** elements that represent outputs coming from this component. |

The following table specifies the attributes of the **PipelineComponentType** complex type.

| Attribute | Description |
|-----------|-------------|
| name | A string value that specifies the name for this **component** element. |
| refId | A string value that specifies a unique **refId** for this object. Elsewhere within the containing executable or within the package, the value of **refId** can be used to refer to this object.<31> |
| componentClassID | An enumeration value that specifies the Class identifier for this component. |
| description | A string value that specifies the description for this **component** element. |
| localeId | An integer value that specifies the language code identifier (LCID) for this component element. |
| usesDispositions | A Boolean value that specifies whether the component has an output path for errors. TRUE specifies that the component has an output path for errors. FALSE specifies that the component does not have an output path for errors. |
| validateExternalMetadata | The **validateExternalMetadata** attribute value is not used by the data transformation processing engine.<32> |

| Attribute | Description |
|---|---|
| version | An integer value that specifies the version of the component. |
| pipelineVersion | An integer value that specifies the version of the task within which a component is designed to execute. |
| contactInfo | A string value that specifies contact information for the developer of a component.<33> |

#### 2.7.1.1.1.1.1 PipelineComponentPropertiesType

The **PipelineComponentPropertiesType** complex type specifies a collection of properties of the component.

The following is the XSD for the **PipelineComponentPropertiesType** complex type.

```
<xs:complexType name="PipelineComponentPropertiesType">
  <xs:sequence>
    <xs:element name="property" form="unqualified"
type="DTS:PipelineComponentPropertyType" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

The following table specifies the elements of the **PipelineComponentPropertiesType** complex type.

| Element | Type definition | Description |
|---|---|---|
| property | PipelineComponentPropertyType | A complex type that specifies a property of a component or of a component input, output, or metadata column. |

#### 2.7.1.1.1.1.1.1 PipelineComponentPropertyType

The **PipelineComponentPropertyType** complex type specifies the structure in which the value for a single property of a component is held. All components do not use the same set of properties, so this general structure is designed to hold the variety of properties that a component can have.

The following is the XSD for the **PipelineComponentPropertyType** complex type.

```
<xs:complexType name="PipelineComponentPropertyType" mixed="true">
  <xs:sequence>
    <xs:element name="arrayElements" type="DTS:PipelineComponentArrayElementsType"
               form="unqualified" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="name" use="required" type="DTS:PipelineComponentPropertyNameEnum"
               form="unqualified"/>
  <xs:attributeGroup ref="DTS:PipelineComponentPropertyAttributeGroup"/>
</xs:complexType>
```

The following table specifies the elements of the **PipelineComponentPropertyType** complex type.

| Element | Type definition | Description |
|---|---|---|
| arrayElements | PipelineComponentArrayElementsType | A complex type that contains a collection of elements |

| Element | Type definition | Description |
|---------|----------------|-------------|
| | | that form an array. The collection constitutes the value for the property. This type is used only if the **isArray** attribute value is TRUE. |

The following table specifies the attributes of the **PipelineComponentPropertyType** complex type.

| Attribute | Description |
|-----------|-------------|
| PipelineComponentPropertyAttributeGroup | An attribute group that contains attributes that are common to all **property** elements for **input** elements, **output** elements, **inputColumn** elements, and **outputColumn** elements. |
| name | A string value that specifies the name of the property. |

#### 2.7.1.1.1.1.1.1.1 PipelineComponentArrayElementsType

The **PipelineComponentArrayElementsType** complex type specifies a collection of component objects for a Pipeline Task executable.

The following is the XSD for the **PipelineComponentArrayElementsType** complex type.

```
<xs:complexType name="PipelineComponentArrayElementsType">
  <xs:sequence>
    <xs:element name="arrayElement" type="DTS:PipelineComponentArrayElementType"
form="unqualified" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="arrayElementCount" type="xs:int" form="unqualified"/>
</xs:complexType>
```

The following table specifies the elements of the **PipelineComponentArrayElementsType** type.

| Element | Type definition | Description |
|---------|----------------|-------------|
| arrayElement | PipelineComponentArrayElementType | A complex type that specifies a single array element that is a member of the array of elements that is the value of a pipeline property. |

#### 2.7.1.1.1.1.1.1.1.1 PipelineComponentArrayElementType

The **PipelineComponentArrayElementType** complex type specifies a collection of component objects for a Pipeline Task executable.

The following is the XSD for the **PipelineComponentArrayElementType** complex type.

```
<xs:complexType name="PipelineComponentArrayElementType">
  <xs:simpleContent>
    <xs:extension base="xs:anySimpleType">
      <xs:attribute name="dataType" type="DTS:PipelineComponentDataTypeEnum"
form="unqualified"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

The following table specifies the attributes of the **PipelineComponentArrayElementType** complex type.

| Attribute | Description |
|-----------|-------------|
| dataType | An enumeration value that specifies the type of the data that is contained in the value for the array element. |

#### 2.7.1.1.1.1.1.1.2    PipelineComponentPropertyNameEnum

The **PipelineComponentPropertyNameEnum** type contains the allowed values for the **Name** attribute on the component element of a Pipeline Task executable. The allowed values for the property element for each task are more restricted than the values that are specified formally by the XSD.

Each name attribute value MUST NOT be used for any component except for the components that are specified in the subsections of this section. Some components allow no properties.

The additional restrictions specified in the subsections of this section MUST be followed for the name attribute on the property element children of the component element.

```
    <xs:simpleType name="PipelineComponentPropertyNameEnum">
      <xs:restriction base="xs:string">
        <!--Transforms-->
        <!--Aggregate Task-->
        <xs:enumeration value="KeyScale"/>
        <xs:enumeration value="Keys"/>
        <xs:enumeration value="CountDistinctScale"/>
        <xs:enumeration value="CountDistinctKeys"/>
        <xs:enumeration value="AutoExtendFactor"/>
        <!--Audit (none)-->
        <!--Cache Transform (none)-->
        <!--Character Map (none)-->
        <!--Conditional Split (none)-->
        <!--Copy Column (none)-->
        <!--Data Conversion (none)-->
        <!--Data Mining Query-->
        <xs:enumeration value="ObjectRef"/>
        <xs:enumeration value="QueryText"/>
        <xs:enumeration value="CatalogName"/>
        <!--Derived Column Query (none)-->
        <!--Export Column Query (none)-->
        <!--Fuzzy Grouping-->
        <xs:enumeration value="MinSimilarity"/>
        <xs:enumeration value="Delimiters"/>
        <xs:enumeration value="MaxMemoryUsage"/>
        <xs:enumeration value="Exhaustive"/>
        <!--Fuzzy Lookup-->
        <xs:enumeration value="MatchIndexOptions"/>
        <xs:enumeration value="MaxMemoryUsage"/>
        <xs:enumeration value="MatchIndexName"/>
        <xs:enumeration value="ReferenceTableName"/>
        <xs:enumeration value="DropExistingMatchIndex"/>
        <xs:enumeration value="CopyReferenceTable"/>
        <xs:enumeration value="MaxOutputMatchesPerInput"/>
        <xs:enumeration value="MinSimilarity"/>
        <xs:enumeration value="Delimiters"/>
        <xs:enumeration value="ReferenceMetadataXml"/>
        <xs:enumeration value="Exhaustive"/>
        <xs:enumeration value="WarmCaches"/>
        <!--Import Column (none)-->
        <!--Lookup-->
        <xs:enumeration value="SqlCommand"/>
        <xs:enumeration value="SqlCommandParam"/>
        <xs:enumeration value="ConnectionType"/>
        <xs:enumeration value="CacheType"/>
        <xs:enumeration value="NoMatchBehavior"/>
```

```
<xs:enumeration value="NoMatchCachePercentage"/>
<xs:enumeration value="MaxMemoryUsage"/>
<xs:enumeration value="MaxMemoryUsage64"/>
<xs:enumeration value="ReferenceMetadataXml"/>
<xs:enumeration value="ParameterMap"/>
<xs:enumeration value="DefaultCodePage"/>
<!--Merge (none)-->
<!--MergeJoin (none)-->
<xs:enumeration value="JoinType"/>
<xs:enumeration value="NumKeyColumns"/>
<xs:enumeration value="TreatNullsAsEqual"/>
<xs:enumeration value="MaxBuffersPerInput"/>
<!--Multicast (none)-->
<!--OLE DB Command-->
<xs:enumeration value="CommandTimeout"/>
<xs:enumeration value="SqlCommand"/>
<xs:enumeration value="DefaultCodePage"/>
<!--Percentage Sampling-->
<xs:enumeration value="SamplingValue"/>
<xs:enumeration value="SamplingSeed"/>
<!--Pivot (none)-->
<!--RowCount-->
<xs:enumeration value="VariableName"/>
<!--Row Sampling-->
<xs:enumeration value="SamplingValue"/>
<xs:enumeration value="SamplingSeed"/>
<!--Script Component-->
<xs:enumeration value="SourceCode"/>
<xs:enumeration value="BinaryCode"/>
<xs:enumeration value="VSTAProjectName"/>
<xs:enumeration value="ScriptLanguage"/>
<xs:enumeration value="ReadOnlyVariables"/>
<xs:enumeration value="ReadWriteVariables"/>
<xs:enumeration value="BreakpointCollection"/>
<xs:enumeration value="UserComponentTypeName"/>
<!--Slowly Changing Dimension-->
<xs:enumeration value="SqlCommand"/>
<xs:enumeration value="CurrentRowWhere"/>
<xs:enumeration value="UpdateChangingAttributeHistory"/>
<xs:enumeration value="FailOnFixedAttributeChange"/>
<xs:enumeration value="InferredMemberIndicator"/>
<xs:enumeration value="EnableInferredMember"/>
<xs:enumeration value="FailOnLookupFailure"/>
<xs:enumeration value="IncomingRowChangeType"/>
<xs:enumeration value="DefaultCodePage"/>
<!--Sort-->
<xs:enumeration value="EliminateDuplicates"/>
<xs:enumeration value="MaximumThreads"/>
<!--Term Extraction-->
<xs:enumeration value="NeedReferenceData"/>
<xs:enumeration value="OutTermTable"/>
<xs:enumeration value="OutTermColumn"/>
<xs:enumeration value="WordOrPhrase"/>
<xs:enumeration value="ScoreType"/>
<xs:enumeration value="FrequencyThreshold"/>
<xs:enumeration value="MaxLengthOfTerm"/>
<xs:enumeration value="IsCaseSensitive"/>
<!--Term Lookup-->
<xs:enumeration value="RefTermTable"/>
<xs:enumeration value="RefTermColumn"/>
<xs:enumeration value="IsCaseSensitive"/>
<!--UnionAll (none)-->
<!--Unpivot (none)-->
<!--Data Destinations-->
<!--ADO.Net Destination-->
<xs:enumeration value="TableOrViewName"/>
<xs:enumeration value="BatchSize"/>
<xs:enumeration value="CommandTimeout"/>
<xs:enumeration value="UserComponentTypeName"/>
<!--Excel Destination-->
```

```xml
<xs:enumeration value="CommandTimeout"/>
<xs:enumeration value="OpenRowset"/>
<xs:enumeration value="OpenRowsetVariable"/>
<xs:enumeration value="SqlCommand"/>
<xs:enumeration value="AccessMode"/>
<xs:enumeration value="FastLoadKeepIdentity"/>
<xs:enumeration value="FastLoadKeepNulls"/>
<xs:enumeration value="FastLoadOptions"/>
<xs:enumeration value="FastLoadMaxInsertCommitSize"/>
<!--Flat File Destination-->
<xs:enumeration value="Overwrite"/>
<xs:enumeration value="Header"/>
<!--OLE DB Destination-->
<xs:enumeration value="CommandTimeout"/>
<xs:enumeration value="OpenRowset"/>
<xs:enumeration value="OpenRowsetVariable"/>
<xs:enumeration value="SqlCommand"/>
<xs:enumeration value="DefaultCodePage"/>
<xs:enumeration value="AlwaysUseDefaultCodePage"/>
<xs:enumeration value="AccessMode"/>
<xs:enumeration value="FastLoadKeepIdentity"/>
<xs:enumeration value="FastLoadKeepNulls"/>
<xs:enumeration value="FastLoadOptions"/>
<xs:enumeration value="FastLoadMaxInsertCommitSize"/>
<!--Raw File Destination-->
<xs:enumeration value="AccessMode"/>
<xs:enumeration value="FileName"/>
<xs:enumeration value="FileNameVariable"/>
<xs:enumeration value="WriteOption"/>
<!--DataReader Destination-->
<xs:enumeration value="ReadTimeout"/>
<xs:enumeration value="DataReader"/>
<xs:enumeration value="FailOnTimeout"/>
<xs:enumeration value="UserComponentTypeName"/>
<!--Data Mining Model Training-->
<xs:enumeration value="ObjectRef"/>
<!--Dimension Processing-->
<xs:enumeration value="ObjectRef"/>
<xs:enumeration value="UseDefaultConfiguration"/>
<xs:enumeration value="KeyErrorLimit"/>
<xs:enumeration value="KeyErrorLogFile"/>
<xs:enumeration value="KeyErrorAction"/>
<xs:enumeration value="KeyErrorLimitAction"/>
<xs:enumeration value="KeyNotFound"/>
<xs:enumeration value="KeyDuplicate"/>
<xs:enumeration value="NullKeyConvertedToUnknown"/>
<xs:enumeration value="NullKeyNotAllowed"/>
<xs:enumeration value="ProcessType"/>
<!--Partition Processing-->
<xs:enumeration value="ObjectRef"/>
<xs:enumeration value="UseDefaultConfiguration"/>
<xs:enumeration value="KeyErrorLimit"/>
<xs:enumeration value="KeyErrorLogFile"/>
<xs:enumeration value="KeyErrorAction"/>
<xs:enumeration value="KeyErrorLimitAction"/>
<xs:enumeration value="KeyNotFound"/>
<xs:enumeration value="KeyDuplicate"/>
<xs:enumeration value="NullKeyConvertedToUnknown"/>
<xs:enumeration value="NullKeyNotAllowed"/>
<xs:enumeration value="ProcessType"/>
<!--RecordSet Destination-->
<xs:enumeration value="VariableName"/>
<!--SQL Server Compact Destination-->
<xs:enumeration value="Table Name"/>
<xs:enumeration value="UserComponentTypeName"/>
<!--SQL Server Destination-->
<xs:enumeration value="DefaultCodePage"/>
<xs:enumeration value="AlwaysUseDefaultCodePage"/>
<xs:enumeration value="BulkInsertTableName"/>
<xs:enumeration value="BulkInsertCheckConstraints"/>
```

```
            <xs:enumeration value="BulkInsertFirstRow"/>
            <xs:enumeration value="BulkInsertFireTriggers"/>
            <xs:enumeration value="BulkInsertKeepIdentity"/>
            <xs:enumeration value="BulkInsertKeepNulls"/>
            <xs:enumeration value="BulkInsertLastRow"/>
            <xs:enumeration value="BulkInsertMaxErrors"/>
            <xs:enumeration value="BulkInsertOrder"/>
            <xs:enumeration value="BulkInsertTablock"/>
            <xs:enumeration value="Timeout"/>
            <xs:enumeration value="MaxInsertCommitSize"/>
            <!--Data Sources-->
            <!--ADO.Net Source-->
            <xs:enumeration value="SqlCommand"/>
            <xs:enumeration value="CommandTimeout"/>
            <xs:enumeration value="AllowImplicitStringConversion"/>
            <xs:enumeration value="TableOrViewName"/>
            <xs:enumeration value="AccessMode"/>
            <xs:enumeration value="UserComponentTypeName"/>
            <!--Excel Source-->
            <xs:enumeration value="CommandTimeout"/>
            <xs:enumeration value="OpenRowset"/>
            <xs:enumeration value="OpenRowsetVariable"/>
            <xs:enumeration value="SqlCommand"/>
            <xs:enumeration value="SqlCommandVariable"/>
            <xs:enumeration value="AccessMode"/>
            <xs:enumeration value="ParameterMapping"/>
            <!--Flat File Source-->
            <xs:enumeration value="RetainNulls"/>
            <xs:enumeration value="FileNameColumnName"/>
            <!--OLEDB Source Task-->
            <xs:enumeration value="CommandTimeout"/>
            <xs:enumeration value="OpenRowset"/>
            <xs:enumeration value="OpenRowsetVariable"/>
            <xs:enumeration value="SqlCommand"/>
            <xs:enumeration value="SqlCommandVariable"/>
            <xs:enumeration value="DefaultCodePage"/>
            <xs:enumeration value="AlwaysUseDefaultCodePage"/>
            <xs:enumeration value="AccessMode"/>
            <xs:enumeration value="ParameterMapping"/>
            <!--Raw File Source-->
            <xs:enumeration value="AccessMode"/>
            <xs:enumeration value="FileName"/>
            <xs:enumeration value="FileNameVariable"/>
            <!--XML Source-->
            <xs:enumeration value="UserComponentTypeName"/>
            <xs:enumeration value="XMLData"/>
            <xs:enumeration value="XMLDataVariable"/>
            <xs:enumeration value="AccessMode"/>
            <xs:enumeration value="XMLSchemaDefinition"/>
            <xs:enumeration value="UseInlineSchema"/>
        </xs:restriction>
    </xs:simpleType>
```

This section presents no tabular explanation of the properties that are specified in the preceding XSD. Tabular explanations are, however, presented in the following subsections when values are allowed.

### 2.7.1.1.1.1.1.1.2.1  Aggregate Task Component

Aggregate Task Component aggregates values by using functions such as sum and average. Aggregate Task Component MUST NOT use any **Name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **Name** attribute values that are allowed for Aggregate Task Component, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| KeyScale | An integer value that specifies the approximate number of groups that is expected by the operation. This value is used to optimize computation. The enumeration values are interpreted as follows:<br><br>0 – The key scale is not used.<br><br>1 – Aggregation writes approximately 500,000 keys.<br><br>2 – Aggregation writes approximately 25,000,000 keys.<br><br>3 – Aggregation writes more than 25,000,000 keys. |
| Keys | An integer value that specifies the number of groups that is expected by the operation. This value is used to optimize computation. If this value is specified, it overrides the value that is specified for the **KeyScale** attribute. |
| CountDistinctScale | An integer value that specifies the approximate number of groups that is expected by a distinct count operation. This value is used to optimize computation. The enumeration values are interpreted as follows:<br><br>0 – The key scale is not used.<br><br>1 – Aggregation writes approximately 500,000 keys.<br><br>2 – Aggregation writes approximately 25,000,000 keys.<br><br>3 – Aggregation writes more than 25,000,000 keys. |
| CountDistinctKeys | An integer value that specifies the number of groups that is expected by a distinct count operation. This value is used to optimize computation. If this value is specified, it overrides the value that is specified for the **CountDistinctKeys** attribute. |
| AutoExtendFactor | An integer value that specifies the percentage amount by which memory is extended during the aggregate operation. |

#### 2.7.1.1.1.1.1.1.2.2 Audit Component

Audit Component allows information about the run-time environment of the package to be inserted into the data flow. Audit Component MUST NOT have any properties.

#### 2.7.1.1.1.1.1.1.2.3 Cache Transform Component

Cache Transform Component copies its inputs to a cache for fast processing. Cache Transform Component MUST NOT have any properties.

#### 2.7.1.1.1.1.1.1.2.4 Character Map Component

Character Map Component processes character strings by applying string functions to the strings, such as converting case and byte order. Character Map Component MUST NOT have any properties.

#### 2.7.1.1.1.1.1.1.2.5 Conditional Split Component

Conditional Split Component splits its input into several outputs, based on conditional logic, to determine which output to use for each input row. Conditional Split Component MUST NOT have any properties.

#### 2.7.1.1.1.1.1.1.2.6 Copy Column Component

Copy Column Component produces a copy of a column on its output. Copy Column Component MUST NOT have any properties.

#### 2.7.1.1.1.1.1.1.2.7 Data Conversion Component

Data Conversion Component converts the data type of a column to a different data type. Data Conversion Component MUST NOT have any properties.

#### 2.7.1.1.1.1.1.2.8   Data Mining Query Component

Data Mining Query Component runs data mining prediction queries. Data Mining Query Component MUST NOT use any **Name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **Name** attribute values that are allowed for Data Mining Query Component, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| ObjectRef | A string value that specifies an XML tag that identifies the mining structure that the task uses. |
| QueryText | A string value that specifies a Data Mining Query Language (DMX) query string. For more information about the DMX language, see [MSDN-DMXR]. |
| CatalogName | A string value that specifies the catalog name in the host database for the data mining model. |

#### 2.7.1.1.1.1.1.2.9   Derived Column Component

Derived Column Component produces new columns by computing them from an expression. Derived Column Component MUST NOT have any properties.

#### 2.7.1.1.1.1.1.2.10   Export Column Component

Export Column Component takes data from the data flow and inserts it into files. Export Column Component MUST NOT have any properties.

#### 2.7.1.1.1.1.1.2.11   Fuzzy Grouping Component

Fuzzy Grouping Component identifies rows of data that are similar enough to be grouped, if the grouping criteria is considered to be fuzzy and not exact. Fuzzy Grouping Component MUST NOT use any **Name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **Name** attribute values that are allowed for Fuzzy Grouping Component, in addition to any restrictions on the element value. All stated restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| MinSimilarity | A single precision number between 0 and 1, inclusive, that specifies the similarity threshold that the task uses to identify duplicates. |
| Delimiters | A string value that specifies the tokens that the task uses as delimiters. |
| MaxMemoryUsage | An integer value that specifies the maximum memory usage for the task in bytes.<34> |
| Exhaustive | A Boolean value that specifies whether each input record is compared to every other input record.<br>TRUE specifies that each input record is compared to every other input record.<br>FALSE specifies that each input record is not compared to every other input record. |

#### 2.7.1.1.1.1.1.2.12   Fuzzy Lookup Component

Fuzzy Lookup Component looks up values in a table and matches them based on similarity and, therefore, does not require an exact match. The match is considered to have been made based on fuzzy criteria. Fuzzy Lookup Component MUST NOT use any **Name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **Name** attribute values that are allowed for Fuzzy Lookup Component, in addition to any restrictions on the element value. All stated restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| MatchIndexOptions | An integer value that specifies how the match index is managed. The values have the following meanings:<br>0 – Reuse existing index.<br>1 – Generate a new index.<br>2 – Generate and persist a new index.<br>3 – Generate and maintain a new index. |
| MaxMemoryUsage | An integer value that specifies the maximum size for the cache table in bytes.<35> |
| MatchIndexName | A string value that specifies the name of the match index. The match index is the table in which the transformation creates and saves the index that it uses. |
| ReferenceTableName | A string value that specifies the name of the lookup table.<36> |
| DropExistingMatchIndex | A Boolean value that specifies whether the match index that is specified in the **MatchIndexName** attribute is deleted when the **MatchIndexOptions** attribute is not set to the value "ReuseExistingIndex". |
| CopyReferenceTable | A Boolean value that specifies whether a copy of the reference table is made for index construction and for subsequent lookups.<br>TRUE specifies that a copy of the reference table is be made.<br>FALSE specifies that a copy of the reference table is not made. |
| MaxOutputMatchesPerInput | An integer value that specifies the maximum number of matches that can be returned for each input row. |
| MinSimilarity | A single precision number between 0 and 1, inclusive, that specifies the similarity threshold that the task uses to identify duplicates. |
| Delimiters | A string value that specifies the tokens that the task uses as delimiters. |
| ReferenceMetadataXml | A string value.<37> |
| Exhaustive | A Boolean value that specifies whether each input record is compared to every other input record.<br>TRUE specifies that each input record is compared to every other input record.<br>FALSE specifies that each input record is not compared to every other input record. |
| WarmCaches | A Boolean value that specifies whether the index and reference table are partially loaded into memory before the operation begins.<br>TRUE specifies that the index and reference table are partially loaded into memory before the operation begins.<br>FALSE specifies that the index and reference table are not partially loaded into memory before the operation begins. |

#### 2.7.1.1.1.1.1.2.13   Import Column Component

Import Column Component reads data from a file and inserts it into a data flow. Import Column Component MUST NOT have any properties.

### 2.7.1.1.1.1.1.2.14   Lookup Component

Lookup Component looks up values in a table and, based on exact matches only, matches those values to data. Lookup Component MUST NOT use any **Name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **Name** attribute values that are allowed for Lookup Component, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| SqlCommand | A string value that specifies an SQL SELECT command string that populates the lookup table. |
| SqlCommandParam | A string value that specifies a parameterized SQL statement that populates the lookup table. |
| ConnectionType | An integer value that specifies the connection type. The allowed values are the following:<br>0 – The connection is OLE DB.<br>1 – The connection is to a cache. |
| CacheType | An integer value that specifies the cache type for the lookup table. The allowed values are interpreted as follows:<br>0 – Full.<br>1 – Partial.<br>2 – No cache. |
| NoMatchBehavior | An integer value that specifies the behavior when rows with no matching value in the reference dataset are encountered in the data flow. The allowed values are interpreted as follows:<br>0 – Rows with no matching entries are treated as errors.<br>1 – Rows with no matching entries are sent to the "no match" output. |
| NoMatchCachePercentage | An integer value between 0 and 100, inclusive, that specifies the maximum percentage of cache memory that can be allocated to data for which a match has not been found. |
| MaxMemoryUsage | An integer value that specifies the maximum memory size in megabytes for the cache on a 32-bit platform. |
| MaxMemoryUsage64 | An integer value that specifies the maximum memory size in megabytes for the cache on a 64-bit platform. |
| ReferenceMetadataXml | A string value.<38> |
| ParameterMap | A string value that specifies the parameters that are mapped to the SQL command. The string is delimited by a semicolon. |
| DefaultCodePage | An integer value that specifies the default code page to use if code page information is not available from the data source. |

### 2.7.1.1.1.1.1.2.15   Merge Component

Merge Component merges, based on values in key columns, two sorted datasets into one output. Merge Component MUST NOT have any properties.

### 2.7.1.1.1.1.1.2.16   MergeJoin Component

MergeJoin Component merges two sorted tables by using a FULL, LEFT, or INNER join. MergeJoin Component MUST NOT use any **Name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **Name** attribute values that are allowed for MergeJoin Component, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| JoinType | An integer value that specifies the type of join to be performed. The interpretation of the enumeration values is as follows.<br>0 – Full join.<br>1 – Left outer join.<br>2 – Inner join. |
| NumKeyColumns | An integer value that specifies the number of key columns used in the join. |
| TreatNullsAsEqual | A Boolean value that specifies whether NULL values are treated as equal for the purpose of a join.<br>TRUE specifies that NULL values are treated as equal.<br>FALSE specifies that NULL values are not treated as equal and are treated as they are normally treated by the database server. |
| MaxBuffersPerInput | An integer value that specifies the maximum number of buffers that is used per input. |

### 2.7.1.1.1.1.1.1.2.17   Multicast Component

Multicast Component duplicates the input onto all of its outputs. Multicast Component MUST NOT have any properties.

### 2.7.1.1.1.1.1.1.2.18   OLE DB Command Component

OLE DB Command Component runs an SQL statement for every row in its data flow. OLE DB Command Component MUST NOT use any **Name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **Name** attribute values that are allowed for OLE DB Command Component, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| CommandTimeout | An integer value that specifies the timeout interval, in seconds, for the command. |
| SqlCommand | A string value that specifies the SQL command string. |
| DefaultCodePage | An integer value that specifies the default code page to use if the code page is not available from the data source. |

### 2.7.1.1.1.1.1.1.2.19   Percentage Sampling Component

Percentage Sampling Component creates a sample data set by randomly selecting a percentage of the transformation input rows. Percentage Sampling Component MUST NOT use any **Name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **Name** attribute values that are allowed for Percentage Sampling Component, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| SamplingValue | An integer value that represents the percentage of the source that is taken as a sample. |
| SamplingSeed | An integer value that specifies the random number generator seed that is used for generating a random sample. |

### 2.7.1.1.1.1.1.2.20   Pivot Component

Pivot Component makes a normalized data set into a less normalized but more compact version by pivoting the input data on a column value. Pivot Component MUST NOT have any properties.

### 2.7.1.1.1.1.1.2.21   RowCount Component

RowCount Component counts rows in the data flow and stores the result in a variable. RowCount Component MUST NOT use any **Name** attribute value on a property element except for those that are listed in the table in this section.

The following table specifies the **Name** attribute values that are allowed for RowCount Component, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| VariableName | A string value that specifies the name of the variable that holds the row count. |

### 2.7.1.1.1.1.1.2.22   Row Sampling Component

Row Sampling Component is used to extract a randomly selected subset of the input data. Row Sampling Component MUST NOT use any **Name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **Name** attribute values that are allowed for Row Sampling Component, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| SamplingValue | An integer value that specifies the number of rows in the sample. |
| SamplingSeed | An integer value that specifies the seed value for the random number generator. |

### 2.7.1.1.1.1.1.2.23   Script Component

Script Component runs custom code that is created in a programming environment within the data flow. Script Component MUST NOT use any **Name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **Name** attribute values that are allowed for the Script Component, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| SourceCode | Specifies the source code for the Script Component. The **property** element value MUST be empty. The **SourceCode** property element MUST contain the **arrayElements** element. |
| BinaryCode | Specifies the binary code for the Script Component. The **property** element value MUST be empty. The **BinaryCode** property element MUST contain the **arrayElements** element. |

| Name attribute value | Description |
|---|---|
| VSTAProjectName | A string value that specifies the project name for the Script Component.<39> |
| ScriptLanguage | A string value from an enumeration that specifies the language of the script project. The valid string values are as follows:<br>CSharp – C# language<br>VisualBasic – Visual Basic language |
| ReadOnlyVariables | A string value that specifies the read-only variables in the package that are available to the Script Component. |
| ReadWriteVariables | A string value that specifies the read-write variables in the package that are available to the Script Component. |
| BreakpointCollection | Specifies a collection of breakpoints for the Script Component. The **property** element value MUST be empty. The **BreakpointCollection** property element MUST contain the **arrayElements** element. |
| UserComponentTypeName | A string value that specifies the fully qualified assembly name for the component. |

### 2.7.1.1.1.1.1.1.2.24  Slowly Changing Dimension Component

Slowly Changing Dimension Component configures the updating of a slowly changing dimension. Slowly Changing Dimension Component MUST NOT use any **Name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **Name** attribute values that are allowed for Slowly Changing Dimension Component, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| SqlCommand | A string value that specifies an SQL statement that is used to create a schema rowset. |
| CurrentRowWhere | A string value that specifies a WHERE clause of an SQL SELECT statement that selects the current row among the rows that have the same key value. |
| UpdateChangingAttributeHistory | A Boolean value that specifies whether historical attribute updates are directed to the transformation output for changing attribute updates. |
| FailOnFixedAttributeChange | A Boolean value that specifies whether the transformation fails when rows where the columns with fixed attributes contain changes or when the lookup in the dimension table fails.<br>TRUE specifies that when fixed attributes change or the lookup fails, the operation fails.<br>FALSE specifies that when fixed attributes change or the lookup fails, the operation identifies a new record. |
| InferredMemberIndicator | A string value that specifies the column name for the inferred member. |
| EnableInferredMember | A Boolean value that specifies whether inferred member updates are detected.<br>TRUE specifies that inferred member updates are detected.<br>FALSE specifies that inferred member updates are not detected. |
| FailOnLookupFailure | A Boolean value that specifies whether lookup failure results in task failure.<br>TRUE specifies that a lookup failure results in a task failure.<br>FALSE specifies that a lookup failure does not result in a task failure. |

| Name attribute value | Description |
|---|---|
| IncomingRowChangeType | An integer value that specifies the incoming row change type. The allowed values are as follows:<br><br>0 – All incoming rows are considered to be new rows.<br><br>1 – The system detects which incoming rows are new. |
| DefaultCodePage | An integer value that specifies the default code page to use if code page information is not available from the data source. |

### 2.7.1.1.1.1.1.2.25  Sort Component

Sort Component sorts data. Sort Component MUST NOT use any **Name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **Name** attribute values that are allowed for Sort Component, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| EliminateDuplicates | A Boolean value that specifies whether to remove duplicate rows from the output.<br><br>TRUE specifies that duplicate rows are removed from the output.<br><br>FALSE specifies that duplicate rows are not removed from the output. |
| MaximumThreads | An integer value that specifies the maximum number of threads to use for sorting.<40> |

### 2.7.1.1.1.1.1.2.26  Term Extraction Component

Term Extraction Component extracts words from text, based on grammatical rules (English language only), to normalize the words. Term Extraction Component MUST NOT use any **Name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **Name** attribute values that are allowed for Term Extraction Component, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| NeedReferenceData | A Boolean value that specifies whether a table of excluded terms is used.<br><br>TRUE specifies that a table of excluded terms is used.<br><br>FALSE specifies that a table of excluded terms is not used. |
| OutTermTable | A string value that specifies the table name that contains the excluded terms. |
| OutTermColumn | A string value that specifies the column name that contains the excluded terms. |
| WordOrPhrase | An integer value that specifies how terms are determined. The valid enumeration values are as follows:<br><br>0 – Words only.<br><br>1 – Noun phrases only.<br><br>2 – Words and noun phrases. |
| ScoreType | An integer value that specifies what type of score is associated with the term.<br><br>0 – The frequency of the term is its score.<br><br>1 – The TFIDF score is the term's score. |

| Name attribute value | Description |
|---|---|
| FrequencyThreshold | An integer value that specifies the number of occurrences of a term before it is extracted. |
| MaxLengthOfTerm | An integer value that specifies the maximum length of terms in characters. It applies only to phrases. |
| IsCaseSensitive | A Boolean value that specifies whether case sensitivity is used when detecting nouns and noun phrases.<br>TRUE specifies that case is used when detecting nouns and noun phrases.<br>FALSE specifies that case is ignored when detecting nouns and noun phrases. |

#### 2.7.1.1.1.1.1.1.2.27  Term Lookup Component

Term Lookup Component looks for a specified set of terms among terms that are extracted by using the same methodology that the Term Extraction Component uses. Term Lookup Component MUST NOT use any **Name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **Name** attribute values that are allowed for Term Lookup Component, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| RefTermTable | A string value that specifies the name of the table that contains the reference terms. |
| RefTermColumn | A string value that specifies the name of the column that contains the reference terms. |
| IsCaseSensitive | A Boolean value that specifies whether case sensitivity is applied to matching input terms with reference terms.<br>TRUE specifies that case is used when matching input terms with reference terms.<br>FALSE specifies that case is ignored when matching input terms with reference terms. |

#### 2.7.1.1.1.1.1.1.2.28  Union All Component

Union All Component combines multiple inputs into one output stream. Union All Component MUST NOT have any properties.

#### 2.7.1.1.1.1.1.1.2.29  Unpivot Component

Unpivot Component makes an unnormalized dataset into a more normalized version by expanding values from multiple columns in a single record into multiple records with the same values in a single column. Unpivot Component MUST NOT have any properties.

#### 2.7.1.1.1.1.1.1.2.30  ADO.NET Destination Component

ADO.NET Destination Component loads data into an ADO.NET-compliant database. ADO.NET Destination Component MUST NOT use any **Name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **Name** attribute values that are allowed for ADO.NET Destination Component, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| TableOrViewName | A string value that specifies the name of the table or view for the task. |

| Name attribute value | Description |
|---|---|
| BatchSize | An integer value that specifies the number of rows that are sent to the server at a time.<41> |
| CommandTimeout | An integer value that specifies the number of seconds that elapse before the command times out.<42> |
| UserComponentTypeName | A string value that specifies the fully qualified Microsoft .NET Framework assembly name for the component. |

### 2.7.1.1.1.1.1.1.2.31   Excel Destination Component

Excel Destination Component writes data to a Microsoft Office Excel workbook. Excel Destination Component MUST NOT use any **Name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **Name** attribute values that are allowed for Excel Destination Component, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| CommandTimeout | An integer value that specifies the number of seconds that elapse before the command times out.<43> |
| OpenRowset | A string value that specifies the name of the table or view that is used. This value is used when the **AccessMode** attribute is set to the value "0" or "3". |
| OpenRowsetVariable | A string value that specifies the name of the variable that contains the name of the table or view that is used. This value is used when the **AccessMode** attribute is set to the value "1" or "4". |
| SqlCommand | A string value that specifies the SQL command that is used. It is used when the **AccessMode** attribute is set to value "2". |
| AccessMode | An integer value that specifies how to access the database at its destination. The allowed values have the following meanings:<br>0 – The name of a table or view is user-specified.<br>1 – The name of a variable that contains the name of a table or view is specified.<br>2 – A user-specified SQL statement is used.<br>3 – Open the rowset with a fast load operation or with a specified table name or view name.<br>4 – Open the rowset with a fast load operation or with the table name or view name specified in a variable. |
| FastLoadKeepIdentity | A Boolean value that specifies whether to copy identity values during a fast load operation. This value is used if the **AccessMode** attribute is set to the value "3" or "4".<br>TRUE specifies that identity values are copied.<br>FALSE specifies that identity values are not copied. |
| FastLoadKeepNulls | A Boolean value that specifies whether to copy NULL values during a fast load operation. This value is used if the AccessMode attribute is set to the value "3" or "4".<br>TRUE specifies that NULL values are to be copied.<br>FALSE specifies that NULL values are not to be copied. |
| FastLoadOptions | A string value that specifies the fast load options that are selected. The string is a comma-separated list of fast load options. For more information about fast |

| Name attribute value | Description |
|---|---|
| | load options, see [MSDN-PBCO].<44> |
| FastLoadMaxInsertCommitSize | An integer value that specifies the commit size, in rows, that is used when the **AccessMode** attribute is set to value "3" or "4". |

#### 2.7.1.1.1.1.1.1.2.32  Flat File Destination Component

Flat File Destination Component writes data to a flat file. Flat File Destination Component MUST NOT use any **Name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **Name** attribute values that are allowed for Flat File Destination Component, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| Header | A string value that contains a string that is inserted into the file header before data is written to the file. |
| Overwrite | A Boolean value that specifies whether to overwrite or to append to an existing destination file with the same name. |
| | TRUE specifies to overwrite a destination file with the same name. |
| | FALSE specifies to append to a destination file with the same name. |

#### 2.7.1.1.1.1.1.1.2.33  OLE DB Destination Component

OLE DB Destination Component loads data into an OLE DB–compliant database. OLE DB Destination Component MUST NOT use any **Name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **Name** attribute values that are allowed for OLE DB Destination Component, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| CommandTimeout | An integer value that specifies the number of seconds that elapse before the command times out.<45> |
| OpenRowset | A string value that specifies the name of the table or view that is used. This value is used when the **AccessMode** attribute is set to the value "0" or "3". |
| OpenRowsetVariable | A string value that specifies the name of the variable that contains the name of the table or view that is used. This value is used when the **AccessMode** attribute is set to the value "1" or "4". |
| SqlCommand | A string value that specifies the SQL command that is used. This value is used when the **AccessMode** attribute is set to value "2". |
| DefaultCodePage | An integer value that specifies the default code page to use if code page information is not available from the data source. |
| AlwaysUseDefaultCodePage | A Boolean value that specifies whether to always use the value for the code page given in the **DefaultCodePage** attribute. |
| | TRUE specifies to always use the value that is given in the **DefaultCodePage** attribute for the code page value. |
| | FALSE specifies to first try to derive the code page value from the data source. If the code page value cannot be derived from the data source, the |

| Name attribute value | Description |
|---|---|
| | **DefaultCodePage** attribute value is used. |
| AccessMode | An integer value that specifies how to access the database at its destination. The allowed values have the following meanings: |
| | 0 – The name of a table or view is user-specified. |
| | 1 – The name of a variable that contains the name of a table or view is specified. |
| | 2 – A user-specified SQL statement is used. |
| | 3 – The rowset is opened by using a fast load operation, with a specified table name or view name. |
| | 4 – The rowset is opened by using a fast load operation, with the table name or view name that is specified in a variable. |
| FastLoadKeepIdentity | A Boolean value that specifies whether to copy identity values during a fast load operation. This value is used if the **AccessMode** attribute is set to the value "3" or "4". |
| | TRUE specifies that identity values are copied. |
| | FALSE specifies that identity values are not copied. |
| FastLoadKeepNulls | A Boolean value that specifies whether to copy NULL values during a fast load operation. This value is used if the **AccessMode** attribute is set to the value "3" or "4". |
| | TRUE specifies that NULL values are to be copied. |
| | FALSE specifies that NULL values are not to be copied. |
| FastLoadOptions | A string value that specifies the fast load options that are selected. The string is a comma-separated list of fast load options. For more information about fast load options, see [MSDN-PBCO].<46> |
| FastLoadMaxInsertCommitSize | An integer value that specifies the commit size, in rows, that is used when **AccessMode** is set to value "3" or "4". |

### 2.7.1.1.1.1.1.1.2.34   DataReader Destination Component

DataReader Destination Component exposes the data in a data flow by using the ADO.NET DataReader interface. The data can then be consumed by other applications. DataReader Destination Component MUST NOT use any **Name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **Name** attribute values that are allowed for DataReader Destination Component, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| ReadTimeout | An integer value that specifies the number of milliseconds that elapse before the operation times out. |
| DataReader | A string value that specifies the class name of the data reader object at the destination. |
| FailOnTimeout | A Boolean value that specifies whether to fail if the value for the **ReadTimeout** attribute is reached. |
| | TRUE specifies to fail if the **ReadTimeout** attribute value is reached. |
| | FALSE specifies not to fail if the **ReadTimeout** attribute value is reached. |
| UserComponentTypeName | A string value that specifies the fully qualified Microsoft .NET Framework assembly name for the component. |

#### 2.7.1.1.1.1.1.1.2.35   Data Mining Model Training Destination Component

Data Mining Model Training Destination Component applies data mining model training algorithms to the data that it receives from the data flow. Data Mining Model Training Destination Component MUST NOT use any **Name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **Name** attribute values that are allowed for Data Mining Model Training Destination Component, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| ObjectRef | A string value that specifies an XML tag that identifies the mining structure that the task uses. |

#### 2.7.1.1.1.1.1.1.2.36   Dimension Processing Destination Component

Dimension Processing Destination Component loads and processes an Analysis Services dimension. Dimension Processing Destination Component MUST NOT use any **Name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **Name** attribute values that are allowed for Dimension Processing Destination Component, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| ObjectRef | A string value that contains a reference to an Analysis Management Objects (AMO) object that represents the dimension after it is looked up.<47> |
| UseDefaultConfiguration | A Boolean value that specifies whether to use the existing default configuration on the server. |
| | TRUE specifies to use the existing default configuration on the server. |
| | FALSE specifies to not use the default configuration on the server. If the value is FALSE, the value that is used is specified in the values of the properties of the component. |
| KeyErrorLimit | An integer value that specifies the number of key errors after which processing fails. This value is used only if the **UseDefaultConfiguration** attribute value is set to FALSE. |
| KeyErrorLogFile | A string value that specifies the file path to the file that is used to log key errors. |

| Name attribute value | Description |
|---|---|
| | This value is used only if the **UseDefaultConfiguration** attribute value is set to FALSE. |
| KeyErrorAction | An integer value that specifies the action to take upon encountering a key processing error. The allowed values are the following:<br><br>0 – Convert the key to the "unknown" value.<br><br>1 – Discard the record.<br><br>This value is used only if the **UseDefaultConfiguration** attribute value is set to FALSE. |
| KeyErrorLimitAction | An integer value that specifies the action to take upon reaching the key error limit. The allowed values are the following:<br><br>0 – Stop processing.<br><br>1 – Continue processing and stop logging.<br><br>This value is used only if the **UseDefaultConfiguration** attribute value is set to FALSE. |
| KeyNotFound | An integer value that specifies the action to take if a key is not found. The allowed values are the following:<br><br>0 – Ignore the error.<br><br>1 – Report the error and continue.<br><br>2 – Report the error and stop processing.<br><br>This value is used only if the **UseDefaultConfiguration** attribute value is set to FALSE. |
| KeyDuplicate | An integer value that specifies the action to take if a duplicate key is found. The allowed values are the following:<br><br>0 – Ignore the error.<br><br>1 – Report the error and continue.<br><br>2 – Report the error and stop processing.<br><br>This value is used only if the **UseDefaultConfiguration** attribute value is set to FALSE. |
| NullKeyConvertedToUnknown | An integer value that specifies the action to take if a null key is found and converted to "unknown". The allowed values are the following:<br><br>0 – Ignore the error.<br><br>1 – Report the error and continue.<br><br>2 – Report the error and stop processing.<br><br>This value is used only if the **UseDefaultConfiguration** attribute value is set to FALSE. |
| NullKeyNotAllowed | An integer value that specifies the action to take if a null key is found that is not allowed. The allowed values are the following:<br><br>0 – Ignore the error.<br><br>1 – Report the error and continue.<br><br>2 – Report the error and stop processing.<br><br>This value is used only if the **UseDefaultConfiguration** attribute value is set to FALSE. |
| ProcessType | An integer value that specifies the type of processing that is performed on the dimension. The allowed values are the following:<br><br>0 – Full processing.<br><br>1 – Processing to add to dimension.<br><br>2 – Update processing.<br><br>For more information about Microsoft SQL Server Analysis Services processing |

| Name attribute value | Description |
|---|---|
| | types, see [MSDN-POS]. |

### 2.7.1.1.1.1.1.1.2.37  Partition Processing Destination Component

Partition Processing Destination Component loads and processes an Analysis Services partition. Partition Processing Destination Component MUST NOT use any **Name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **Name** attribute values that are allowed for Partition Processing Destination Component, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| ObjectRef | A string value that contains a reference to an Analysis Management Objects (AMO) object that represents the partition after it is looked up.<48> |
| UseDefaultConfiguration | A Boolean value that specifies whether to use the existing default configuration on the server.<br>TRUE specifies to use the existing default configuration on the server.<br>FALSE specifies to not use the default configuration on the server. If the value is FALSE, the values that are used are specified in the values of the properties of the component. |
| KeyErrorLimit | An integer value that specifies the number of key errors after which processing fails. This value is used only if the **UseDefaultConfiguration** attribute value is set to FALSE. |
| KeyErrorLogFile | A string value that specifies the file path to the file that is used to log key errors. This value is used only if the **UseDefaultConfiguration** attribute value is set to FALSE. |
| KeyErrorAction | An integer value that specifies the action to take upon encountering a key processing error. The allowed values are the following:<br>0 – Convert the key to the "unknown" value.<br>1 – Discard the record.<br>This value is used only if the **UseDefaultConfiguration** attribute value is set to FALSE. |
| KeyErrorLimitAction | An integer value that specifies the action to take upon reaching the key error limit. The allowed values are the following:<br>0 – Stop processing.<br>1 – Continue processing and stop logging.<br>This value is used only if the **UseDefaultConfiguration** attribute value is set to FALSE. |
| KeyNotFound | An integer value that specifies the action to take if a key is not found. The allowed values are the following:<br>0 – Ignore the error.<br>1 – Report the error and continue.<br>2 – Report the error and stop processing.<br>This value is used only if the **UseDefaultConfiguration** attribute value is set to FALSE. |
| KeyDuplicate | An integer value that specifies the action to take if a duplicate key is found. The allowed values are the following:<br>0 – Ignore the error. |

| Name attribute value | Description |
|---|---|
| | 1 – Report the error and continue. |
| | 2 – Report the error and stop processing. |
| | This value is used only if the **UseDefaultConfiguration** attribute value is set to FALSE. |
| NullKeyConvertedToUnknown | An integer value that specifies the action to take if a null key is found and converted to "unknown". The allowed values are the following: |
| | 0 – Ignore the error. |
| | 1 – Report the error and continue. |
| | 2 – Report the error and stop processing. |
| | This value is used only if the **UseDefaultConfiguration** attribute value is set to FALSE. |
| NullKeyNotAllowed | An integer value that specifies the action to take if a null key is found that is not allowed. The allowed values are the following: |
| | 0 – Ignore the error. |
| | 1 – Report the error and continue. |
| | 2 – Report the error and stop processing. |
| | This value is used only if the **UseDefaultConfiguration** attribute value is set to FALSE. |
| ProcessType | An integer value that specifies the type of processing that is performed on the dimension. The allowed values are the following: |
| | 0 – Full processing. |
| | 1 – Processing to add to dimension. |
| | 2 – Update processing. |
| | For more information about Analysis Services processing types, see [MSDN-POS]. |

### 2.7.1.1.1.1.1.2.38  Raw File Destination Component

Raw File Destination Component writes raw data to a file in a format that is native to the file type. Raw File Destination Component MUST NOT use any **Name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **Name** attribute values that are allowed for Raw File Destination Component, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| AccessMode | An integer value that specifies whether the file name is specified in the XML or is specified as the value of a variable. The allowed values are the following: |
| | 0 – The file name is specified in the XML. |
| | 1 – The file name is specified as the value of a variable. |
| FileName | A string value that specifies the name of the file for output. This value is used only if the value of the **AccessMode** attribute is "0". |
| FileNameVariable | A string value that specifies the name of the variable that contains the file name for the output file. This value applies only if the value for the **AccessMode** attribute is equal to "1". |
| WriteOption | An integer value that specifies the action to take for the file write operation. The allowed values are the following: |
| | 0 – Always create the file. |
| | 1 – Create the file one time only. |

| Name attribute value | Description |
|---|---|
| | 2 – Append to an existing file.<br>3 – Truncate the file and append. |

### 2.7.1.1.1.1.1.2.39   RecordSet Destination Component

RecordSet Destination Component creates and populates an in-memory ADO recordset. RecordSet Destination Component MUST NOT use any **Name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **Name** attribute values that are allowed for RecordSet Destination Component, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| VariableName | A string value that specifies the name of the variable that holds the ADO recordset. |

### 2.7.1.1.1.1.1.2.40   SQL Server Compact Destination Component

SQL Server Compact Destination Component loads data into a SQL Server Compact database. SQL Server Compact Destination Component MUST NOT use any **Name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **Name** attribute values that are allowed for SQL Server Compact Destination Component, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| TableName | A string value that specifies the name of the destination table in the SQL Server Compact database. |
| UserComponentTypeName | A string value that specifies the fully qualified Microsoft .NET Framework assembly name for the component. |

### 2.7.1.1.1.1.1.2.41   SQL Server Destination Component

SQL Server Destination Component connects to a local SQL Server database and bulk loads data into SQL Server tables and views on a server that is local to where the package is executing. SQL Server Destination Component MUST NOT use any **Name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **Name** attribute values that are allowed for SQL Server Destination Component, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| DefaultCodePage | An integer value that specifies the default code page to use if code page information is not available from the data source. |
| AlwaysUseDefaultCodePage | A Boolean value that specifies whether to always use the value for the code page that is given in the **DefaultCodePage** attribute.<br>TRUE specifies to always use the value that is given in the **DefaultCodePage** attribute for the code page value.<br>FALSE specifies to first try to derive the code page value from the data source. If the code page value cannot be derived from the data source, the **DefaultCodePage** attribute value is used. |

| Name attribute value | Description |
|---|---|
| BulkInsertTableName | A string value that specifies the table name to use for the bulk insert operation. |
| BulkInsertCheckConstraints | A Boolean value that specifies whether constraints on the table are checked during the bulk insert operation.<br>TRUE specifies that constraints are checked.<br>FALSE specifies that constraints are not checked. |
| BulkInsertFirstRow | An integer value that specifies the row in the table at which insertion begins.<49> |
| BulkInsertFireTriggers | A Boolean value that specifies whether insert triggers that are specified for the table are fired during the bulk insert operation.<br>TRUE specifies that triggers are fired during the bulk insert operation.<br>FALSE specifies that triggers are not fired during the bulk insert operation. |
| BulkInsertKeepIdentity | A Boolean value that specifies whether to keep data values from the data file for values for identity columns.<br>TRUE specifies that the values that are present in the data file are kept for identity columns.<br>FALSE specifies that the values that are present in the data file for identity columns are not kept. |
| BulkInsertKeepNulls | A Boolean value that specifies whether null values in the input are retained.<br>TRUE specifies that null values in the input are retained.<br>FALSE specifies that a default value is used if a null value is encountered in the input. |
| BulkInsertLastRow | An integer value that specifies the row in the data file at which insertion ends.<50> |
| BulkInsertMaxErrors | An integer value that specifies the maximum number of errors that are allowed. |
| BulkInsertOrder | A string value that specifies the columns that are used for sorting. |
| BulkInsertTablock | A Boolean value that specifies whether a table level lock is acquired for the bulk insert operation.<br>TRUE specifies that a table level lock is acquired for the bulk insert operation.<br>FALSE specifies that a table level lock is not acquired for the bulk insert operation. |
| Timeout | An integer value that specifies the number of seconds that the destination waits for records to insert before the operation times out.<51> |
| MaxInsertCommitSize | An integer value that specifies the maximum number of rows to insert in a single batch.<52> |

### 2.7.1.1.1.1.1.1.2.42   ADO.NET Source Component

ADO.NET Source Component accepts data from an ADO.NET-compliant database as its source of data. ADO.NET Source Component MUST NOT use any **Name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **Name** attribute values that are allowed for ADO.Net Source Component, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| SqlCommand | A string value that specifies an SQL SELECT command string that populates the lookup table. |
| CommandTimeout | An integer value that specifies the number of seconds before the command times |

| Name attribute value | Description |
|---|---|
| | out.<53> |
| AllowImplicitStringConversion | A Boolean value that specifies whether external metadata types are converted to the type that the destination requires.<54> |
| | TRUE specifies that external types are converted to strings. |
| | FALSE specifies that no error is generated if there is a mismatch between the external metadata type and a column that requires a string. |
| TableOrViewName | A string value that specifies the name of the table or view for the task. |
| AccessMode | An integer value that specifies the access mode to the ADO.NET data. The allowed values are the following: |
| | 0 – Table or view. |
| | 2 – SQL command. |
| UserComponentTypeName | A string value that specifies the fully qualified Microsoft .NET Framework assembly name for the component. |

### 2.7.1.1.1.1.1.1.2.43  Excel Source Component

Excel Source Component accepts data from an Office Excel workbook as its source of data. Excel Source Component MUST NOT use any **Name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **Name** attribute values that are allowed for Excel Source Component, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| CommandTimeout | An integer value that specifies the number of seconds that elapse before the command times out.<55> |
| OpenRowset | A string value that specifies the name of the table or view that is used. This value is used when the **AccessMode** attribute is set to the value "0" or "3". |
| OpenRowsetVariable | A string value that specifies the name of the variable that contains the name of the table or view that is used. This value is used when the **AccessMode** attribute is set to the value "1" or "4". |
| SqlCommand | A string value that specifies the SQL command that is used. This value is used when the **AccessMode** attribute is set to the value "2". |
| SqlCommandVariable | A string value that specifies the name of the variable that contains the SQL command that is executed. |
| AccessMode | An integer value that specifies how to access the data in the Office Excel sheet. The allowed values have the following meanings: |
| | 0 – Use "open rowset" to open the Office Excel data. |
| | 1 – Use "open rowset" from the value that is contained in a variable to open the Office Excel data. |
| | 2 – Use an SQL command to open the Office Excel data. |
| | 3 – Use an SQL command that is contained in a variable to open the Office Excel data. |
| ParameterMapping | A string value that specifies the mapping of SQL parameters to values that are contained in variables. |

### 2.7.1.1.1.1.1.1.2.44  Flat File Source Component

Flat File Source Component accepts data from a flat file as its source of data. Flat File Source Component MUST NOT use any **Name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **Name** attribute values that are allowed for Flat File Source Component, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| RetainNulls | A Boolean value that specifies whether to retain NULL values that are obtained from the source during the pipeline processing.<br>TRUE specifies that NULL values are retained during pipeline processing.<br>FALSE specifies that NULL values are not retained. |
| FileNameColumnName | A string value that specifies the name of the output column that contains the file name for the flat file source file. |

### 2.7.1.1.1.1.1.1.2.45  OLE DB Source Component

OLE DB Source Component accepts data from an OLE DB–compliant database as its source of data. OLE DB Source Component MUST NOT use any **Name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **Name** attribute values that are allowed for the OLE DB Source Component, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| CommandTimeout | An integer value that specifies the number of seconds that elapse before the command times out.<56> |
| OpenRowset | A string value that specifies the name of the table or view that is used. This value is used when the **AccessMode** attribute is set to the value "0" or "3". |
| OpenRowsetVariable | A string value that specifies the name of the variable that contains the name of the table or view that is used. This value is used when the **AccessMode** attribute is set to the value "1" or "4". |
| SqlCommand | A string value that specifies the SQL command that is used. This value is used when the **AccessMode** attribute is set to value "2". |
| DefaultCodePage | An integer value that specifies the default code page to use if code page information is not available from the data source. |
| AlwaysUseDefaultCodePage | A Boolean value that specifies whether to always use the value for the code page given in the **DefaultCodePage** attribute.<br>TRUE specifies to always use the value given in the **DefaultCodePage** attribute for the code page value.<br>FALSE specifies to first try to derive the code page value from the data source. If the code page value cannot be derived from the data source, the **DefaultCodePage** attribute value is used. |
| AccessMode | An integer value that specifies how to access the data in the Office Excel sheet. The allowed values have the following meanings.<br>0 – Use "open rowset" to open the Office Excel data.<br>1 – Use "open rowset" from the value that is contained in a variable to open the Office Excel data.<br>2 – Use an SQL command to open the Office Excel data.<br>3 – Use an SQL command that is contained in a variable to open the Office Excel |

| Name attribute value | Description |
|---|---|
| | data. |
| ParameterMapping | A string value that specifies the mapping of SQL parameters to values that are contained in variables. |

### 2.7.1.1.1.1.1.2.46   Raw File Source Component

Raw File Source Component accepts data from a raw file as its source of data. Raw File Source Component MUST NOT use any **Name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **Name** attribute values that are allowed for Raw File Source Component, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| AccessMode | An integer value that specifies whether the file is specified in the XML or is specified as the value of a variable. The allowed values are the following:<br>0 – The file name is specified in the package file.<br>1 – The file name is specified as the value of a variable. |
| FileName | A string value that specifies the name of the file for output. This value is used only if the value of the **AccessMode** attribute is "0". |
| FileNameVariable | A string value that specifies the name of the variable that contains the file name for the output file. This value applies only if the value for the **AccessMode** attribute is equal to "1". |

### 2.7.1.1.1.1.1.2.47   XML Source Component

XML Source Component accepts data from an XML file as its source of data. XML Source Component MUST NOT use any **Name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **Name** attribute values that are allowed for XML Source Component, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| UserComponentTypeName | A string value that specifies the fully qualified Microsoft .NET Framework assembly name for the component. |
| XMLData | A string value that specifies the file name or variable name that contains the XML. The content of the string is dependent upon the value of the **AccessMode** attribute. |
| XMLDataVariable | A string value that specifies the name of the variable that contains the XML data. |
| AccessMode | An integer value that specifies the mode of access of the XML. The allowed values are the following.<br>0 – XML file.<br>1 - XML file from the value of a variable.<br>2 – XML data from the value of a variable. |
| XMLSchemaDefinition | A string value that contains an inline XML Schema definition. This value is used if the **UseInlineSchema** attribute value is set to TRUE. |
| UseInlineSchema | A Boolean value that specifies whether an inline XML Schema definition is used. |

| Name attribute value | Description |
|---|---|
| | TRUE specifies that an inline XML Schema definition is used. |
| | FALSE specifies that an inline XML Schema definition is not used. |

### 2.7.1.1.1.1.1.1.3 PipelineComponentDataTypeEnum

The **PipelineComponentDataTypeEnum** type contains the enumeration values for the **dataType** attribute.

The following is the XSD for the **PipelineComponentDataTypeEnum** type.

```
<xs:simpleType name="PipelineComponentDataTypeEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="System.Boolean"/>
    <xs:enumeration value="System.Byte"/>
    <xs:enumeration value="System.SByte"/>
    <xs:enumeration value="System.Decimal"/>
    <xs:enumeration value="System.Double"/>
    <xs:enumeration value="System.Single"/>
    <xs:enumeration value="System.Int32"/>
    <xs:enumeration value="System.UInt32"/>
    <xs:enumeration value="System.Int64"/>
    <xs:enumeration value="System.UInt64"/>
    <xs:enumeration value="System.Int16"/>
    <xs:enumeration value="System.UInt16"/>
    <xs:enumeration value="System.String"/>
    <xs:enumeration value="System.Null"/>
    <xs:enumeration value="System.DateTime"/>
    <xs:enumeration value="System.Object"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **PipelineComponentDataTypeEnum** type.

| Enumeration value | Description |
|---|---|
| System.Boolean | The value is a Boolean. |
| System.Byte | The value is a byte. |
| System.SByte | The value is an 8-bit signed integer. |
| System.Decimal | The value is a decimal quantity. |
| System.Double | The value is a double precision real number. |
| System.Single | The value is a single precision real number. |
| System.Int32 | The value is a 32-bit integer. |
| System.UInt32 | The value is a 32-bit unsigned integer. |
| System.Int64 | The value is a 64-bit integer. |
| System.UInt64 | The value is a 64-bit unsigned integer. |
| System.Int16 | The value is a 16-bit integer. |

| Enumeration value | Description |
|---|---|
| System.UInt16 | The value is a 16-bit unsigned integer. |
| System.String | The value is a string. |
| System.Null | The value is null. |
| System.DateTime | The value is a date/time value. |
| System.Object | The value is an object. |

### 2.7.1.1.1.1.1.1.4    PipelineComponentStateEnum

The **PipelineComponentStateEnum** type contains the enumeration values for the **state** attribute.

The following is the XSD for the **PipelineComponentStateEnum** type.

```
<xs:simpleType name="PipelineComponentStateEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="cdata"/>
    <xs:enumeration value="escaped"/>
    <xs:enumeration value="default"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **PipelineComponentStateEnum** type.

| Enumeration value | Description |
|---|---|
| cdata | Serialize the property value as a CDATA XSD section. |
| escaped | Serialize the property value as text, with escape sequence as needed. |
| default | Serialize the property value in the default fashion. |

### 2.7.1.1.1.1.1.1.5    PipelineComponentExpressionTypeEnum

The **PipelineComponentExpressionTypeEnum** type contains the enumeration values for the **expressionType** attribute.

The following is the XSD for the **PipelineComponentExpressionTypeEnum** type.

```
<xs:simpleType name="PipelineComponentExpressionTypeEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="None"/>
    <xs:enumeration value="Notify"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the
**PipelineComponentExpressionTypeEnum** type.

| Enumeration value | Description |
|---|---|
| None | The value of the property cannot be specified with an expression. |
| Notify | The value of the property can be specified with a property expression. |

### 2.7.1.1.1.1.2 PipelineComponentConnectionsType

The **PipelineComponentConnectionsType** complex type specifies a collection of connection objects for a component.

The following is the XSD for the **PipelineComponentConnectionsType** complex type.

```
<xs:complexType name="PipelineComponentConnectionsType">
  <xs:sequence>
    <xs:element name="connection" form="unqualified"
                type="DTS:PipelineComponentConnectionType"
                minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

The following table specifies the elements of the **PipelineComponentConnectionsType** complex type.

| Element | Type definition | Description |
|---|---|---|
| **connection** | PipelineComponentConnectionType | A complex type that specifies a reference to a connection manager that is available to this component. |

### 2.7.1.1.1.1.2.1 PipelineComponentConnectionType

The **PipelineComponentConnectionType** complex type specifies a collection of connection objects for a component.

The following is the XSD for the **PipelineComponentConnectionType** complex type.

```
<xs:complexType name="PipelineComponentConnectionType">
  <xs:sequence/>
  <xs:attribute name="refId" type="xs:string" use="required"
                form="unqualified"/>
  <xs:attribute name="name" type="xs:string" use="optional"
                default="" form="unqualified"/>
  <xs:attribute name="description" type="xs:string"
                use="optional" default="" form="unqualified"/>
  <xs:attribute name="connectionManagerID" type="xs:string"
                use="optional" default="" form="unqualified"/>
</xs:complexType>
```

The following table specifies the attributes of the **PipelineComponentConnectionType** complex type.

| Attribute | Description |
|---|---|
| refId | A string value that specifies a unique **refId** for this object. Elsewhere within the |

| Attribute | Description |
|---|---|
|  | containing executable or within the package, the value of **refId** can be used to refer to this object.<57> |
| name | A string value that specifies the name for this component connection element. |
| description | A string value that specifies the description for this component connection element. |
| connectionManagerID | A UUID value that specifies the **DTSID** attribute of the connection manager that is used by this component connection. |

### 2.7.1.1.1.1.3 PipelineComponentInputsType

The **PipelineComponentInputsType** complex type specifies a collection of output objects for a component.

The following is the XSD for the **PipelineComponentInputsType** complex type.

```
<xs:complexType name="PipelineComponentInputsType">
  <xs:sequence>
    <xs:element name="input" form="unqualified"
                type="DTS:PipelineComponentInputType"
                minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

The following table specifies the elements of the **PipelineComponentInputsType** complex type.

| Element | Type definition | Description |
|---|---|---|
| input | PipelineComponentInputType | A complex type that specifies a reference to an input that is available to this component. |

### 2.7.1.1.1.1.3.1 PipelineComponentInputType

The **PipelineComponentInputType** complex type specifies the structure in which the descriptor for a single input to a component is held. All inputs are not identical, so this general structure is designed to hold the variety of inputs that a component can have.

The following is the XSD for the **PipelineComponentInputType** complex type.

```
<xs:complexType name="PipelineComponentInputType">
  <xs:sequence>
    <xs:element name="properties"
                type="DTS:PipelineComponentInputPropertiesType"
                minOccurs="0" form="unqualified"/>
    <xs:element name="inputColumns"
                type="DTS:PipelineComponentInputColumnsType"
                minOccurs="0" form="unqualified"/>
    <xs:element name="externalMetadataColumns"
                type="DTS:PipelineComponentMetadataColumnsType"
                minOccurs="0" form="unqualified"/>
  </xs:sequence>
  <xs:attributeGroup
    ref="DTS:PipelineComponentInputOutputElementAttributeGroup"/>
```

```
    </xs:complexType>
```

The following table specifies the elements of the **PipelineComponentInputType** complex type.

| Element | Type definition | Description |
|---|---|---|
| properties | PipelineComponentInputPropertiesType | A complex type that specifies a collection of properties for an **input** element. |
| inputColumns | PipelineComponentInputColumnsType | A complex type that specifies a collection of input columns that are associated with the **input** element. |
| externalMetadataColumns | PipelineComponentMetadataColumnsType | A complex type that specifies a collection of external metadata columns that are associated with the **input** element. |

The following table specifies the attribute groups of the **PipelineComponentInputType** complex type.

| Attribute group | Description |
|---|---|
| PipelineComponentInputOutputElementAttributeGroup | An attribute group specification that specifies attributes that are common for the **input** element and for the **output** element of the **PipelineComponentInputType** type. |

### 2.7.1.1.1.1.3.1.1 PipelineComponentInputColumnsType

The **PipelineComponentInputColumnsType** complex type specifies a collection of column objects for an **input** element that is part of a Pipeline component.

The following is the XSD for the **PipelineComponentInputColumnsType** complex type.

```
<xs:complexType name="PipelineComponentInputColumnsType">
  <xs:sequence>
    <xs:element name="inputColumn"
                type="DTS:PipelineComponentInputColumnType"
                minOccurs="0" maxOccurs="unbounded" form="unqualified"/>
  </xs:sequence>
</xs:complexType>
```

The following table specifies the elements of the **PipelineComponentInputColumnsType** complex type.

| Element | Type definition | Description |
|---|---|---|
| inputColumn | PipelineComponentInputColumnType | A complex type that specifies a single column within a pipeline input. |

### 2.7.1.1.1.1.3.1.2 PipelineComponentInputColumnType

The **PipelineComponentInputColumnType** complex type specifies a single column that is a member of a collection of input columns to a Pipeline component.

The following is the XSD for the **PipelineComponentInputColumnType** complex type.

```
<xs:complexType name="PipelineComponentInputColumnType">
  <xs:sequence>
    <xs:element name="properties" type="DTS:PipelineComponentInputColumnPropertiesType"
                form="unqualified" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attributeGroup ref="DTS:PipelineComponentAllColumnBaseAttributeGroup"/>
  <xs:attributeGroup ref="DTS:PipelineComponentIOColumnBaseAttributeGroup"/>
  <xs:attribute name="usageType"  form="unqualified"
                use="optional" default="readOnly"
                type="DTS:PipelineComponentUsageTypeEnum"/>
  <xs:attribute name="cachedName" type="xs:string"
                use="optional" default="" form="unqualified"/>
  <xs:attribute name="cachedSortKeyPosition" type="xs:int"
                use="optional" default="0" form="unqualified"/>
  <xs:attribute name="cachedPrecision" type="xs:int" use="optional"
                default="0" form="unqualified"/>
  <xs:attribute name="cachedScale" type="xs:int" use="optional"
                default="0" form="unqualified"/>
  <xs:attribute name="cachedLength" type="xs:int" use="optional"
                default="0" form="unqualified"/>
  <xs:attribute name="cachedDataType"
                type="DTS:PipelineComponentColumnDataTypeEnum"
                use="optional" default="empty" form="unqualified"/>
  <xs:attribute name="cachedCodepage" type="xs:int" use="optional"
                default="0" form="unqualified"/>
  <xs:attribute name="cachedComparisonFlags" type="xs:int"
                form="unqualified" use="optional" default="0" />
</xs:complexType>
```

The following table specifies the elements of the **PipelineComponentInputColumnType** complex type.

| Element | Type definition | Description |
|---------|----------------|-------------|
| properties | PipelineComponentInputColumnPropertiesType | A complex type that specifies a collection of properties of an input column. |

The following table specifies the attributes and attribute groups of the **PipelineComponentInputColumnType** complex type.

| Attribute group or attribute | Description |
|------------------------------|-------------|
| PipelineComponentAllColumnBaseAttributeGroup | An attribute group specification that specifies attributes that are common to all types of columns for Pipeline components. |
| PipelineComponentIOColumnBaseAttributeGroup | An attribute group specification that specifies attributes that are common to all input and output columns for Pipeline components. |
| usageType | An enumeration value that specifies the type of use available for the input column. |
| cachedName | A string value that specifies the name for this column element. |
| cachedSortKeyPosition | An integer value that specifies the position of this column among the sort keys. |
| cachedPrecision | An integer value that specifies the total digits in a numeric |

| Attribute group or attribute | Description |
|---|---|
| | data type. |
| cachedScale | An integer value that specifies the number of digits to the right of the decimal point. |
| cachedLength | An integer value that specifies the length of the column in characters. |
| cachedDataType | An enumeration value that specifies the type of the data in the column. |
| cachedCodepage | An integer value that specifies the numeric value of the ANSI code page that is used for strings for this column. |
| cachedComparisonFlags | An integer value that is a bitmask that specifies which options are chosen for comparison of values in the column. The interpretation of the bits in this integer is as follows: <br><br> 0x00000001 – Ignore case. <br><br> 0x00000002 – Ignore nonspace characters. <br><br> 0x00000004 – Ignore symbols. <br><br> 0x00000010 – Linguistically appropriate ignore case. <br><br> 0x00000020 – Linguistically appropriate ignore nonspace characters. <br><br> 0x00010000 – Ignore Kana type. <br><br> 0x00020000 - Ignore width. <br><br> 0x08000000 – Use linguistic rules for casing. |

### 2.7.1.1.1.1.3.1.3    PipelineComponentInputColumnPropertiesType

The **PipelineComponentInputColumnPropertiesType** complex type specifies a collection of properties for an **inputColumn** element that is part of a Pipeline component.

The following is the XSD for the **PipelineComponentInputColumnPropertiesType** complex type.

```
<xs:complexType name="PipelineComponentInputColumnPropertiesType">
  <xs:sequence>
    <xs:element name="property" form="unqualified"
                type="DTS:PipelineComponentInputColumnPropertyType"
                minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

The following table specifies the elements of the **PipelineComponentInputColumnPropertiesType** complex type.

| Element | Type definition | Description |
|---|---|---|
| property | PipelineComponentInputColumnPropertyType | A complex type that specifies a single property within a collection of input column properties for an **inputColumn** element of a Pipeline component. |

### 2.7.1.1.1.1.3.1.3.1    PipelineComponentInputColumnPropertyType

The **PipelineComponentInputColumnPropertyType** complex type specifies a property of an **inputColumn** element.

The following is the XSD for the **PipelineComponentInputColumnPropertyType** complex type.

```
<xs:complexType name="PipelineComponentInputColumnPropertyType">
  <xs:simpleContent>
    <xs:extension base="xs:anySimpleType">
      <xs:attributeGroup ref="DTS:PipelineComponentPropertyAttributeGroup"/>
      <xs:attribute name="name" form="unqualified" use="required"
          type="DTS:PipelineComponentInputColumnPropertyNameEnum"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

The following table specifies the attributes and attribute groups of the **PipelineComponentInputColumnPropertyType** complex type.

| Attribute group or attribute | Description |
|---|---|
| PipelineComponentPropertyAttributeGroup | An attribute group that contains attributes that are common to all **property** elements for **input** elements, **output** elements, **inputColumn** elements, and **outputColumn** elements. |
| name | A string value that specifies the name of the property. |

### 2.7.1.1.1.1.3.1.3.1.1  PipelineComponentInputColumnPropertyNameEnum

The **PipelineComponentInputColumnPropertyNameEnum** type contains the allowed values for the **name** attribute on the **inputColumn** element of a Pipeline component. The allowed values for the **property** element for each task are more restricted than the values that are specified formally by the XSD.

Each **name** attribute value MUST NOT be used for any component except for the components that are specified in the subsections to this section. Some components allow no properties.

The additional restrictions specified in the subsections to this section MUST be followed for the **name** attribute on the **property** element children of the **inputColumn** element.

The following is the XSD for the **PipelineComponentInputColumnPropertyNameEnum** type.

```
<xs:simpleType name="PipelineComponentInputColumnPropertyNameEnum">
  <xs:restriction base="xs:string">
    <!--Transforms-->
    <!--Aggregate Task-->
    <!--Audit (none)-->
    <!--Cache Transform-->
    <xs:enumeration value="CacheColumnName"/>
    <!--Character Map-->
    <xs:enumeration value="MapFlags"/>
    <!--Conditional Split (none)-->
    <!--Copy Column (none)-->
    <!--Data Conversion (none)-->
    <!--Data Mining Query(none)-->
    <!--Derived Column Query-->
    <xs:enumeration value="Expression"/>
    <xs:enumeration value="FriendlyExpression"/>
    <!--Export Column-->
    <xs:enumeration value="FileDataColumnID"/>
    <xs:enumeration value="WriteBOM"/>
    <xs:enumeration value="ForceTruncate"/>
    <xs:enumeration value="AllowAppend"/>
    <!--Fuzzy Grouping-->
    <xs:enumeration value="ToBeCleaned"/>
    <xs:enumeration value="ExactFuzzy"/>
    <xs:enumeration value="LeadingTrailingNumeralsSignificant"/>
```

```
        <xs:enumeration value="MinSimilarity"/>
        <xs:enumeration value="FuzzyComparisonFlags"/>
        <!--Fuzzy Lookup-->
        <xs:enumeration value="JoinType"/>
        <xs:enumeration value="ExactFuzzy"/>
        <xs:enumeration value="JoinToReferenceColumn"/>
        <xs:enumeration value="FuzzyComparisonFlags"/>
        <xs:enumeration value="FuzzyComparisonFlagsEx"/>
        <!--Import Column-->
        <xs:enumeration value="FileDataColumnID"/>
        <xs:enumeration value="ExpectBOM"/>
        <!--Lookup-->
        <xs:enumeration value="JoinToReferenceColumn"/>
        <xs:enumeration value="CopyFromReferenceColumn"/>
        <!--Merge-->
        <xs:enumeration value="OutputColumnLineageID"/>
        <!--MergeJoin (none)-->
        <!--Multicast (none)-->
        <!--OLE DB Command (none)-->
        <!--Percentage Sampling (none)-->
        <!--Pivot-->
        <xs:enumeration value="PivotUsage"/>
        <!--RowCount(none)-->
        <!--Row Sampling(none)-->
        <!--Script Component(none)-->
        <!--Slowly Changing Dimension-->
        <xs:enumeration value="ColumnType"/>
        <!--Sort-->
        <xs:enumeration value="NewComparisonFlags"/>
        <xs:enumeration value="NewSortKeyPosition"/>
        <!--Term Extraction(none)-->
        <!--Term Lookup-->
        <xs:enumeration value="InputColumnType"/>
        <!--UnionAll-->
        <xs:enumeration value="OutputColumnLineageID"/>
        <!--Unpivot-->
        <xs:enumeration value="DestinationColumn"/>
        <xs:enumeration value="PivotKeyValue"/>
        <!--Data Destinations-->
        <!--ADO.Net Destination(none)-->
        <!--Excel Destination(none)-->
        <!--Flat File Destination(none)-->
        <!--OLE DB Destination(none)-->
        <!--Raw File Destination(none)-->
        <!--DataReader Destination(none)-->
        <!--Data Mining Model Training(none)-->
        <!--Dimension Processing(none)-->
        <!--Partition Processing(none)-->
        <!--RecordSet Destination(none)-->
        <!--SQL Server Compact Destination(none)-->
        <!--SQL Server Destination(none)-->
    </xs:restriction>
  </xs:simpleType>
```

#### 2.7.1.1.1.1.3.1.3.1.1.1  Aggregate Task Component inputColumn Element

The Aggregate Task Component MUST NOT have any properties on the **inputColumn** element.

#### 2.7.1.1.1.1.3.1.3.1.1.2  Audit Component inputColumn Element

The Audit Component MUST NOT have any properties on the **inputColumn** element.

#### 2.7.1.1.1.1.3.1.3.1.1.3  Cache Transform Component inputColumn Element

The Cache Transform Component **inputColumn** element MUST NOT use any **name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **name** attribute values that are allowed for Cache Transform Component on the **inputColumn** element, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| CacheColumnName | A string value that specifies the name of a column that is associated with an input column. |

#### 2.7.1.1.1.1.3.1.3.1.1.4 Character Map Component inputColumn Element

The Character Map Component **inputColumn** element MUST NOT use any **name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **name** attribute values that are allowed for Character Map Component on the **inputColumn** element, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| MapFlags | An integer value that specifies the string operation that is performed on the column. The allowed values are the following: <br> 0 – Map to lower case. <br> 1 – Map to upper case. <br> 2 – Perform byte reversal. <br> 3 – Map to Hiragana. <br> 4 – Map to Katakana. <br> 5 – Map to half width. <br> 6 – Map to full width. <br> 7 – Perform linguistic casing. <br> 8 – Map to Simplified Chinese. <br> 9 – Map to Traditional Chinese. |

#### 2.7.1.1.1.1.3.1.3.1.1.5 Conditional Split Component inputColumn Element

The Conditional Split Component MUST NOT have any properties on the **inputColumn** element.

#### 2.7.1.1.1.1.3.1.3.1.1.6 Copy Column Component inputColumn Element

The Copy Column Component MUST NOT have any properties on the **inputColumn** element.

#### 2.7.1.1.1.1.3.1.3.1.1.7 Data Conversion Component inputColumn Element

The Data Conversion Component MUST NOT have any properties on the **inputColumn** element.

#### 2.7.1.1.1.1.3.1.3.1.1.8 Data Mining Query Component inputColumn Element

The Data Mining Query Component MUST NOT have any properties on the **inputColumn** element.

#### 2.7.1.1.1.1.3.1.3.1.1.9 Derived Column Component inputColumn Element

The Derived Column Component **inputColumn** element MUST NOT use any **name** attribute value on a property element except for those that are listed in the table in this section.

The following table specifies the **name** attribute values that are allowed for Derived Column Component on the **inputColumn** element, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| Expression | A string value that specifies the expression by which the column is derived. |
| FriendlyExpression | A string value that specifies a conditional expression by which the column is derived. The **FriendlyExpression** attribute uses column names in the expression syntax, while the **Expression** attribute uses only column lineage identifiers. |

### 2.7.1.1.1.1.3.1.3.1.1.10 Export Column Component inputColumn Element

The Export Column Component **inputColumn** element MUST NOT use any **name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **name** attribute values that are allowed for the Export Column Component on the **inputColumn** element, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| FileDataColumnID | An integer value that specifies the identifier for the column that contains the data that is inserted by the task. |
| WriteBOM | A Boolean value that specifies whether a byte order mark (BOM) is inserted into the file. <br> TRUE specifies that a BOM is inserted into the file. <br> FALSE specifies that a BOM is not inserted into the file. |
| ForceTruncate | A Boolean value that specifies whether a truncation is forced before a write operation on an existing file. <br> TRUE specifies that a truncation is forced before a write operation on an existing file. <br> FALSE specifies that a truncation is not forced before a write operation on an existing file. |
| AllowAppend | A Boolean value that specifies whether the transformation is allowed to append data to an existing file. <br> TRUE specifies that the transformation is allowed to append data to an existing file. <br> FALSE specifies that the transformation is not allowed to append data to an existing file. |

### 2.7.1.1.1.1.3.1.3.1.1.11 Fuzzy Grouping Component inputColumn Element

The Fuzzy Grouping Component **inputColumn** element MUST NOT use any **name** attribute value on a property element except for those that are listed in the table in this section.

The following table specifies the **name** attribute values that are allowed for the Fuzzy Grouping Component on the **inputColumn** element, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| ToBeCleaned | A Boolean value that specifies whether this is a column to be grouped by. <br> TRUE specifies that this is a column to be grouped by. <br> FALSE specifies that this is not a column to be grouped by. |
| ExactFuzzy | An integer value that specifies whether the fuzzy match is exact or fuzzy. The allowed values are the following: <br> 1 – Exact. <br> 2 – Fuzzy. |

| Name attribute value | Description |
|---|---|
| LeadingTrailingNumeralsSignificant | An integer value that specifies how to treat numerals. The allowed values are as follows:<br>0 – Numerals are not significant.<br>1 – Leading numerals are significant.<br>2 – Trailing numerals are significant.<br>3 – Both leading and trailing numerals are significant. |
| MinSimilarity | A double precision value between 0 and 1, inclusive, that specifies the similarity threshold to join on the column. Only rows that are greater than this value qualify as matches. |
| FuzzyComparisonFlags | An integer value that is a bitmask that specifies the comparison flags for the sort operation. The bit values are as follows:<br>0x00000001 - Ignore case.<br>0x00000002 - Ignore nonspace characters.<br>0x00000004 - Ignore symbols.<br>0x00000100 - Sort punctuation as symbols.<br>0x00001000 - Ignore Kana type.<br>0x00002000 - Ignore character width. |

#### 2.7.1.1.1.1.3.1.3.1.1.12   Fuzzy Lookup Component inputColumn Element

The Fuzzy Lookup Component **inputColumn** element MUST NOT use any **name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **name** attribute values that are allowed for the Fuzzy Lookup Component on the **inputColumn** element, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| ToBeCleaned | A Boolean value that specifies whether this is a column to be grouped by.<br>TRUE specifies that this is a column to be grouped by.<br>FALSE specifies that this is not a column to be grouped by. |
| ExactFuzzy | An integer value that specifies whether the fuzzy match is exact or fuzzy. The allowed values are the following:<br>1 – Exact<br>2 – Fuzzy |
| LeadingTrailingNumeralsSignificant | An integer value that specifies how to treat numerals. The allowed values are as follows:<br>0 – Numerals are not significant.<br>1 – Leading numerals are significant.<br>2 – Trailing numerals are significant.<br>3 – Both leading and trailing numerals are significant. |
| MinSimilarity | A double precision value between 0 and 1, inclusive, that specifies the similarity threshold to join on the column. Only rows that are greater than this value qualify as matches. |
| FuzzyComparisonFlags | An integer value that is a bitmask that specifies the comparison flags for the sort operation. The bit values are as follows:<br>0x00000001 - Ignore case.<br>0x00000002 - Ignore nonspace characters. |

| Name attribute value | Description |
|---|---|
| | 0x00000004 - Ignore symbols. |
| | 0x00000100 - Sort punctuation as symbols. |
| | 0x00001000 - Ignore Kana type. |
| | 0x00002000 - Ignore character width. |

#### 2.7.1.1.1.1.3.1.3.1.1.13  Import Column Component inputColumn Element

The Import Column Component **inputColumn** element MUST NOT use any **name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **name** attribute values that are allowed for the Import Column Component on the **inputColumn** element, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| FileDataColumnID | An integer value that specifies the identifier for the column that contains the data that is inserted into the data flow. |
| ExpectBOM | A Boolean value that specifies whether a byte order mark (BOM) is expected in the column. |
| | TRUE specifies that a BOM is expected in the column. |
| | FALSE specifies that a BOM is not expected in the column. |

#### 2.7.1.1.1.1.3.1.3.1.1.14  Lookup Component inputColumn Element

The Lookup Component **inputColumn** element MUST NOT use any **name** attribute value on a property element except for those that are listed in the table in this section.

The following table specifies the **name** attribute values that are allowed for the Lookup Component on the **inputColumn** element, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| JoinToReferenceColumn | A string value that specifies the name of the column in the reference table upon which to join. |
| CopyFromReferenceColumn | A string value that specifies the name of the column in the reference table from which a column is copied. |

#### 2.7.1.1.1.1.3.1.3.1.1.15  Merge Component inputColumn Element

The Merge Component MUST NOT have any properties on the **inputColumn** element.

#### 2.7.1.1.1.1.3.1.3.1.1.16  MergeJoin Component inputColumn Element

The MergeJoin Component MUST NOT have any properties on the **inputColumn** element.

#### 2.7.1.1.1.1.3.1.3.1.1.17  Multicast Component inputColumn Element

The Multicast Component MUST NOT have any properties on the **inputColumn** element.

#### 2.7.1.1.1.1.3.1.3.1.1.18  OLE DB Command Component inputColumn Element

The OLE DB Command Component MUST NOT have any properties on the **inputColumn** element.

### 2.7.1.1.1.3.1.3.1.1.19 Percentage Sampling Component inputColumn Element

The Percentage Sampling Component MUST NOT have any properties on the **inputColumn** element.

### 2.7.1.1.1.3.1.3.1.1.20 Pivot Component inputColumn Element

The Pivot Component **inputColumn** element MUST NOT use any **name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **name** attribute values that are allowed for the Pivot Component on the **inputColumn** element, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| **PivotUsage** | An integer value that specifies the role of a column for pivoting. The allowed values are the following:<br>0 – The column is not pivoted.<br>1 – The column provides row keys. All input rows with the same value are combined into one output row.<br>2 – The column is a pivot column. At least one column is created for each value in this column.<br>3 – The values from this column are placed in one of the cells created by the pivot. |

### 2.7.1.1.1.3.1.3.1.1.21 RowCount Component inputColumn Element

The RowCount Component MUST NOT have any properties on the **inputColumn** element.

### 2.7.1.1.1.3.1.3.1.1.22 Row Sampling Component inputColumn Element

The Row Sampling Component MUST NOT have any properties on the **inputColumn** element.

### 2.7.1.1.1.3.1.3.1.1.23 Script Component inputColumn Element

The Script Component MUST NOT have any properties on the **inputColumn** element.

### 2.7.1.1.1.3.1.3.1.1.24 Slowly Changing Dimension Component inputColumn Element

The Slowly Changing Dimension Component **inputColumn** element MUST NOT use any **name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **name** attribute values that are allowed for the Slowly Changing Dimension Component on the **inputColumn** element, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| **ColumnType** | An integer value that specifies the update type for the column.<br>0 – Unknown.<br>1 – Key column.<br>2 – Changing attribute.<br>3 – Historical attribute.<br>4 – Fixed attribute. |

### 2.7.1.1.1.3.1.3.1.1.25 Sort Component inputColumn Element

The Sort Component **inputColumn** element MUST NOT use any **name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **name** attribute values that are allowed for the Sort Component on the **inputColumn** element, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| **NewComparisonFlags** | An integer value that is a bitmask that specifies the comparison flags for the sort operation. The bit values are as follows:<br>0x00000001 - Ignore case.<br>0x00000002 - Ignore nonspace characters.<br>0x00000004 - Ignore symbols.<br>0x00000100 - Sort punctuation as symbols.<br>0x00001000 - Ignore Kana type.<br>0x00002000 - Ignore character width. |
| **NewSortKeyPosition** | An integer value that specifies the sort order of the column. |

#### 2.7.1.1.1.1.3.1.3.1.1.26   Term Extraction Component inputColumn Element

The Term Extraction Component MUST NOT have any properties on the **inputColumn** element.

#### 2.7.1.1.1.1.3.1.3.1.1.27   Term Lookup Component inputColumn Element

The Term Lookup Component **inputColumn** element MUST NOT use any **name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **name** attribute values that are allowed for the Term Lookup Component on the **inputColumn** element, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| InputColumnType | An integer value that specifies the input column type. The allowed values are the following:<br>0 – Passthrough column.<br>1 – Lookup column.<br>2 – Passthrough and lookup column. |

#### 2.7.1.1.1.1.3.1.3.1.1.28   Union All Component inputColumn Element

The Union All Component **inputColumn** element MUST NOT use any **name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **name** attribute values that are allowed for the Union All Component on the **inputColumn** element, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| OutputColumnLineageID | An integer value that specifies the lineage identifier of the corresponding output column. |

#### 2.7.1.1.1.1.3.1.3.1.1.29   Unpivot Component inputColumn Element

The Unpivot Component **inputColumn** element MUST NOT use any **name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **name** attribute values that are allowed for the Unpivot Component on the **inputColumn** element, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| DestinationColumn | An integer value that specifies the lineage identifier of the corresponding output column.<58> |
| PivotKeyValue | A string value that specifies one of the pivot key values that is copied to an output column. |

**2.7.1.1.1.1.3.1.3.1.1.30   ADO.NET Destination Component inputColumn Element**

The ADO.Net Destination Component MUST NOT have any properties on the **inputColumn** element.

**2.7.1.1.1.1.3.1.3.1.1.31   Excel Destination Component inputColumn Element**

The Excel Destination Component MUST NOT have any properties on the **inputColumn** element.

**2.7.1.1.1.1.3.1.3.1.1.32   Flat File Destination Component inputColumn Element**

The Flat File Destination Component MUST NOT have any properties on the **inputColumn** element.

**2.7.1.1.1.1.3.1.3.1.1.33   OLE DB Destination Component inputColumn Element**

The OLE DB Destination Component MUST NOT have any properties on the **inputColumn** element.

**2.7.1.1.1.1.3.1.3.1.1.34   DataReader Destination Component inputColumn Element**

The DataReader Destination Component MUST NOT have any properties on the **inputColumn** element.

**2.7.1.1.1.1.3.1.3.1.1.35   Data Mining Model Training Destination Component inputColumn Element**

The Data Mining Model Training Destination Component MUST NOT have any properties on the **inputColumn** element.

**2.7.1.1.1.1.3.1.3.1.1.36   Dimension Processing Destination Component inputColumn Element**

The Dimension Processing Destination Component MUST NOT have any properties on the **inputColumn** element.

**2.7.1.1.1.1.3.1.3.1.1.37   Partition Processing Destination Component inputColumn Element**

The Partition Processing Destination Component MUST NOT have any properties on the **inputColumn** element.

**2.7.1.1.1.1.3.1.3.1.1.38   Raw File Destination Component inputColumn Element**

The Raw File Destination Component MUST NOT have any properties on the **inputColumn** element.

**2.7.1.1.1.1.3.1.3.1.1.39   RecordSet Destination Component inputColumn Element**

The RecordSet Destination Component MUST NOT have any properties on the **inputColumn** element.

**2.7.1.1.1.1.3.1.3.1.1.40   SQL Server Compact Destination Component inputColumn Element**

The SQL Server Compact Destination Component MUST NOT have any properties on the **inputColumn** element.

#### 2.7.1.1.1.1.3.1.3.1.1.41   SQL Server Destination Component inputColumn Element

The SQL Server Destination Component MUST NOT have any properties on the **inputColumn** element.

#### 2.7.1.1.1.1.3.1.3.2   PipelineComponentUsageTypeEnum

The **PipelineComponentUsageTypeEnum** type contains the enumeration values for the **usageType** attribute.

The following is the XSD for the **PipelineComponentUsageTypeEnum** type.

```
<xs:simpleType name="PipelineComponentUsageTypeEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="readOnly"/>
    <xs:enumeration value="readWrite"/>
    <xs:enumeration value="Ignored"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **PipelineComponentUsageTypeEnum** type.

| Enumeration value | Description |
|---|---|
| readOnly | The column is read-only. |
| readWrite | The column is available for read operations and for write operations. |
| Ignored | The column is ignored. |

#### 2.7.1.1.1.1.3.1.4      PipelineComponentInputPropertiesType

The **PipelineComponentInputPropertiesType** complex type specifies a collection of properties for an **input** element that is part of a Pipeline component.

The following is the XSD for the **PipelineComponentInputPropertiesType** complex type.

```
<xs:complexType name="PipelineComponentInputPropertiesType">
  <xs:sequence>
    <xs:element name="property" form="unqualified"
                type="DTS:PipelineComponentInputPropertyType"
                minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

The following table specifies the elements of the **PipelineComponentInputPropertiesType** complex type.

| Element | Type definition | Description |
|---|---|---|
| property | PipelineComponentInputPropertyType | A complex type that specifies a single property within a collection of input properties for an **input** element of a Pipeline component. |

### 2.7.1.1.1.1.3.1.5 PipelineComponentInputPropertyType

The **PipelineComponentInputPropertyType** complex type specifies the structure in which the value for a single property of an input of a component is held. All inputs do not use the same set of properties, so this general structure is designed to hold the variety of properties that an input can have.

The following is the XSD for the **PipelineComponentInputPropertyType** complex type.

```
<xs:complexType name="PipelineComponentInputPropertyType"
                mixed="true">
  <xs:sequence>
    <xs:element name="arrayElements"
                type="DTS:PipelineComponentArrayElementsType"
                form="unqualified" minOccurs="0"/>
  </xs:sequence>
  <xs:attributeGroup
    ref="DTS:PipelineComponentPropertyAttributeGroup"/>
  <xs:attribute name="name" use="required"
                type="DTS:PipelineComponentInputPropertyNameEnum"
                form="unqualified"/>
</xs:complexType>
```

The following table specifies the elements of the **PipelineComponentInputPropertyType** complex type.

| Element | Type definition | Description |
|---|---|---|
| arrayElements | PipelineComponentArrayElementsType | A complex type that contains a collection of elements that form an array. The collection constitutes the value for the property. This element is used only if the value of the **isArray** attribute is TRUE. |

The following table specifies the attribute groups and the attributes of the **PipelineComponentInputPropertyType** complex type.

| Attribute group or attribute | Description |
|---|---|
| PipelineComponentPropertyAttributeGroup | An attribute group that contains attributes that are common to all **property** elements for **input** elements, **output** elements, **inputColumn** elements, and **outputColumn** elements. |
| name | A string value that specifies the name of the property. |

### 2.7.1.1.1.1.3.1.6 PipelineComponentInputPropertyNameEnum

The **PipelineComponentInputPropertyNameEnum** type contains the allowed values for the **name** attribute on the **input** element of a Pipeline component. The allowed values for the **property** element for each task are more restricted than the values that are specified formally by the XSD.

No tasks allow any property elements on the **input** element. Therefore, the **PipelineComponentInputPropertyNameEnum** type does not contain any enumerated values. It is specified as a type as the **property** element itself is valid, and custom components can add valid property names for the name attribute. For more information about the addition of properties by custom programmers, see Custom Packages.

All tasks MUST NOT have any properties on the **input** element.

The following is the XSD for the **PipelineComponentInputPropertyNameEnum** type.

```
<xs:simpleType name="PipelineComponentInputPropertyNameEnum">
  <xs:restriction base="xs:string">
  </xs:restriction>
<!--None of the components have properties for the Input element-->
</xs:simpleType>
```

### 2.7.1.1.1.1.4 PipelineComponentOutputsType

The **PipelineComponentOutputsType** complex type specifies a collection of output objects for a component.

The following is the XSD for the **PipelineComponentOutputsType** complex type.

```
<xs:complexType name="PipelineComponentOutputsType">
  <xs:sequence>
    <xs:element name="output" form="unqualified"
                type="DTS:PipelineComponentOutputType"
                minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

The following table specifies the elements of the **PipelineComponentOutputsType** complex type.

| Element | Type definition | Description |
|---------|-----------------|-------------|
| output | PipelineComponentOutputType | A complex type that specifies a reference to an output that is created by this component and that is available to other components. |

### 2.7.1.1.1.1.4.1 PipelineComponentOutputType

The **PipelineComponentOutputType** complex type specifies the structure in which the descriptor for a single output from a component is held. All outputs are not identical, so this general structure is designed to hold the variety of outputs that a component can have.

The following is the XSD for the **PipelineComponentOutputType** complex type.

```
<xs:complexType name="PipelineComponentOutputType">
  <xs:sequence>
    <xs:element name="properties"
                type="DTS:PipelineComponentOutputPropertiesType"
                minOccurs="0" form="unqualified"/>
    <xs:element name="outputColumns"
                type="DTS:PipelineComponentOutputColumnsType"
                minOccurs="0" form="unqualified"/>
    <xs:element name="externalMetadataColumns"
                type="DTS:PipelineComponentMetadataColumnsType"
```

```
                  minOccurs="0" form="unqualified"/>
        </xs:sequence>
        <xs:attributeGroup
          ref="DTS:PipelineComponentInputOutputElementAttributeGroup"/>
        <xs:attribute name="exclusionGroup" type="xs:int"
                   use="optional" default="0" form="unqualified"/>
        <xs:attribute name="synchronousInputId" type="xs:string"
                   use="optional" default="0" form="unqualified"/>
        <xs:attribute name="deleteOutputOnPathDetached" type="xs:boolean"
                   use="optional" default="false" form="unqualified"/>
        <xs:attribute name="isErrorOut" type="xs:boolean"
                   use="optional" default="false" form="unqualified"/>
        <xs:attribute name="isSorted" type="xs:boolean" use="optional"
                   default="false" form="unqualified"/>
    </xs:complexType>
```

The following table specifies the elements of the **PipelineComponentOutputType** complex type.

| Element | Type definition | Description |
|---------|----------------|-------------|
| properties | PipelineComponentPropertiesType | A complex type that specifies a collection of properties that are associated with the **output** element. |
| outputColumns | PipelineComponentOutputColumnsType | A complex type that specifies a collection of output columns that are associated with the **output** element. |
| externalMetadataColumns | PipelineComponentMetadataColumnsType | A complex type that specifies a collection of external metadata columns that are associated with the **output** element. |

The following table specifies the attribute groups and the attributes of the **PipelineComponentOutputType** complex type.

| Attribute group or attribute | Description |
|------------------------------|-------------|
| PipelineComponentInputOutputElementAttributeGroup | An attribute group specification that specifies attributes that are common for the **input** element and for the **output** element of the **PipelineComponentOutputType** type. |
| exclusionGroup | An integer value that specifies a hint to the data flow engine about whether copies of flowing data need to be made for multiple outputs. If multiple outputs share the same **exclusionGroup** value, the engine guarantees that input rows are output to only one of those outputs. If the **exclusionGroup** value is "0" or if the **exclusionGroup** attribute for outputs is unequal, no such guarantee is made. |
| synchronousInputId | A string value that specifies the **refId** attribute of an input that is synchronous with this output. |
| deleteOutputOnPathDetached | A Boolean value that determines whether the data flow engine deletes the output when it is detached from a path. TRUE specifies that the output is deleted when it is detached from a path. FALSE specifies that the output is not deleted when it is detached from a path. |

| Attribute group or attribute | Description |
|---|---|
| isErrorOut | A Boolean value that specifies whether the output is an error output.<br><br>TRUE specifies that the output is an error output.<br><br>FALSE specifies that the output is not an error output. |
| isSorted | A Boolean value that specifies whether the output is sorted.<br><br>TRUE specifies that the output is sorted.<br><br>FALSE specifies that the output is not sorted. |

#### 2.7.1.1.1.1.4.1.1　PipelineComponentOutputColumnsType

The **PipelineComponentOutputColumnsType** complex type specifies a collection of column objects for an **output** element that is part of a Pipeline component.

The following is the XSD for the **PipelineComponentOutputColumnsType** complex type.

```
<xs:complexType name="PipelineComponentOutputColumnsType">
  <xs:sequence>
    <xs:element name="outputColumn"
                type="DTS:PipelineComponentOutputColumnType"
          minOccurs="0" maxOccurs="unbounded" form="unqualified"/>
  </xs:sequence>
</xs:complexType>
```

The following table specifies the elements of the **PipelineComponentOutputColumnsType** complex type.

| Element | Type definition | Description |
|---|---|---|
| outputColumn | PipelineComponentOutputColumnType | A complex type that specifies a single column within a collection of output columns for a Pipeline component. |

#### 2.7.1.1.1.1.4.1.1.1　PipelineComponentOutputColumnType

The **PipelineComponentOutputColumnType** complex type specifies a single column that is a member of a collection of output columns in a Pipeline component.

The following is the XSD for the **PipelineComponentOutputColumnType** complex type.

```
<xs:complexType name="PipelineComponentOutputColumnType">
  <xs:sequence>
    <xs:element name="properties"
                type="DTS:PipelineComponentOutputColumnPropertiesType"
                form="unqualified" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attributeGroup ref="DTS:PipelineComponentAllColumnBaseAttributeGroup"/>
  <xs:attributeGroup ref="DTS:PipelineComponentIOColumnBaseAttributeGroup"/>
  <xs:attributeGroup ref="DTS:PipelineComponentColumnExtendedAttributeGroup"/>
  <xs:attribute name="sortKeyPosition" type="xs:int"
                use="optional" default="0" form="unqualified"/>
  <xs:attribute name="comparisonFlags" type="xs:int"
                form="unqualified" use="optional" default="0" />
  <xs:attribute name="specialFlags" form="unqualified"
                use="optional" default="0">
    <xs:simpleType>
```

```
            <xs:restriction base="xs:int">
               <xs:minInclusive value="0"/>
               <xs:maxInclusive value="2"/>
            </xs:restriction>
         </xs:simpleType>
      </xs:attribute>
   </xs:complexType>
```

The following table specifies the elements of the **PipelineComponentOutputColumnType** complex type.

| Element | Type definition | Description |
|---------|-----------------|-------------|
| properties | PipelineComponentOutputColumnPropertiesType | A complex type that specifies a collection of properties for an output column. |

The following table specifies the attributes and attribute groups of the **PipelineComponentOutputColumnType** complex type.

| Attribute group or attribute | Description |
|------------------------------|-------------|
| PipelineComponentAllColumnBaseAttributeGroup | An attribute group specification that specifies attributes that are common to all types of columns for Pipeline components. |
| PipelineComponentIOColumnBaseAttributeGroup | An attribute group specification that specifies attributes that are common to all input and output columns for Pipeline components. |
| PipelineComponentColumnExtendedAttributeGroup | An attribute group specification that specifies additional attributes present for the **PipelineComponentOutputColumnType** complex type. |
| sortKeyPosition | An integer value that specifies the position of this column among the sort keys. |
| comparisonFlags | An integer value that is a bitmask that specifies which options are chosen for comparison of values in the column. The interpretation of the bits in this integer is as follows: <br>0x00000001 – Ignore case. <br>0x00000002 – Ignore nonspace characters. <br>0x00000004 – Ignore symbols. <br>0x00000010 – Linguistically appropriate ignore case. <br>0x00000020 – Linguistically appropriate ignore nonspace characters. <br>0x00010000 – Ignore Kana type. <br>0x00020000 - Ignore width. <br>0x08000000 – Use linguistic rules for casing. |
| specialFlags | An integer enumeration value that specifies whether the column contains special information. The allowed values are the following: <br>0 – No special information is contained in the column. <br>1 – The data in the column contains error codes. <br>2 – The data in the column contains the lineage identifier of the column in which an error occurred. |

#### 2.7.1.1.1.1.4.1.1.2  PipelineComponentOutputColumnPropertiesType

The **PipelineComponentOutputColumnPropertiesType** complex type specifies a collection of properties for an **outputColumn** element that is part of a Pipeline component.

The following is the XSD for the **PipelineComponentOutputColumnPropertiesType** complex type.

```
<xs:complexType name="PipelineComponentOutputColumnPropertiesType">
  <xs:sequence>
    <xs:element name="property" form="unqualified"
                type="DTS:PipelineComponentOutputColumnPropertyType"
                minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

The following table specifies the elements of the **PipelineComponentOutputColumnPropertiesType** complex type.

| Element | Type definition | Description |
|---------|-----------------|-------------|
| property | PipelineComponentOutputColumnPropertyType | A complex type that specifies a single property within a collection of input column properties for an **outputColumn** element of a Pipeline component. |

#### 2.7.1.1.1.1.4.1.1.2.1　PipelineComponentOutputColumnPropertyType

The **PipelineComponentOutputColumnPropertyType** complex type specifies a property for an **outputColumn** element.

The following is the XSD for the **PipelineComponentOutputColumnPropertyType** complex type.

```
<xs:complexType name="PipelineComponentOutputColumnPropertyType">
  <xs:simpleContent>
    <xs:extension base="xs:anySimpleType">
      <xs:attributeGroup ref="DTS:PipelineComponentPropertyAttributeGroup"/>
      <xs:attribute name="name" form="unqualified" use="required"
type="DTS:PipelineComponentOutputColumnPropertyNameEnum"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

The following table specifies the attribute groups of the **PipelineComponentOutputColumnPropertyType** complex type.

| Attribute group | Description |
|-----------------|-------------|
| PipelineComponentPropertyAttributeGroup | An attribute group that contains attributes that are common to all **property** elements for **input** elements, **output** elements, **inputColumn** elements, and **outputColumn** elements. |
| name | An enumeration value that specifies the name of the property. |

#### 2.7.1.1.1.1.4.1.1.2.1.1　PipelineComponentOutputColumnPropertyNameEnum

The **PipelineComponentOutputColumnPropertyNameEnum** type contains the allowed values for the **name** attribute on the **outputColumn** element of a Pipeline component. The allowed values for the **property** element for each task are more restricted than the values that are specified formally by the XSD.

Each **name** attribute value MUST NOT be used for any component except for the components that are specified in the subsections to this section. Some components allow no properties.

The additional restrictions specified in the subsections to this section MUST be followed for the **name** attribute on the **property** element children of the **outputColumn** element.

```
<xs:simpleType name="PipelineComponentOutputColumnPropertyNameEnum">
  <xs:restriction base="xs:string">
    <!--Transforms-->
    <!--Aggregate Task-->
    <xs:enumeration value="AggregationColumnId"/>
    <xs:enumeration value="AggregationType"/>
    <xs:enumeration value="IsBig"/>
    <xs:enumeration value="AggregationComparisonFlags"/>
    <xs:enumeration value="CountDistinctKey"/>
    <xs:enumeration value="CountDistinctScale"/>
    <!--Audit-->
    <xs:enumeration value="LineageItemSelected"/>
    <!--Cache Transform (none)-->
    <!--Character Map (none)-->
    <!--Conditional Split (none)-->
    <!--Copy Column (none)-->
    <xs:enumeration value="copyColumnId"/>
    <!--Data Conversion (none)-->
    <xs:enumeration value="SourceInputColumnLineageID"/>
    <xs:enumeration value="FastParse"/>
    <!--Data Mining Query-->
    <!--Derived Column Query (none)-->
    <xs:enumeration value="Expression"/>
    <xs:enumeration value="FriendlyExpression"/>
    <!--Export Column Query (none)-->
    <!--Fuzzy Grouping-->
    <xs:enumeration value="ColumnType"/>
    <xs:enumeration value="InputId"/>
    <!--Fuzzy Lookup-->
    <xs:enumeration value="CopyFromReferenceColumn"/>
    <xs:enumeration value="SourceInputColumnLineageID"/>
    <xs:enumeration value="ColumnType"/>
    <!--Import Column (none)-->
    <!--Lookup-->
    <xs:enumeration value="CopyFromReferenceColumn"/>
    <!--Merge (none)-->
    <!--MergeJoin-->
    <xs:enumeration value="InputColumnID"/>
    <!--Multicast (none)-->
    <!--OLE DB Command (none)-->
    <!--Percentage Sampling (none)-->
    <!--Pivot-->
    <xs:enumeration value="SourceColumn"/>
    <xs:enumeration value="PivotKeyValue"/>
    <!--RowCount (none)-->
    <!--Row Sampling-->
    <xs:enumeration value="InputColumnLineageID"/>
    <!--Script Component (none)-->
    <!--Slowly Changing Dimension (none)-->
    <!--Sort-->
    <xs:enumeration value="SortColumnId"/>
    <!--Term Extraction(none)-->
    <!--Term Lookup-->
    <xs:enumeration value="CustomLineageID"/>
    <!--UnionAll (none)-->
    <!--Unpivot (none)-->
    <xs:enumeration value="PivotKey"/>
    <!--Data Sources-->
    <!--ADO.Net Source (none)-->
    <!--Excel Source(none)-->
    <!--Flat File Source-->
    <xs:enumeration value="FastParse"/>
    <xs:enumeration value="UseBinaryFormat"/>
    <!--OLEDB Source Task(none)-->
    <!--Raw File Source(none)-->
    <!--XML Source(none)-->
```

```
        </xs:restriction>
    </xs:simpleType>
```

#### 2.7.1.1.1.1.4.1.1.2.1.1.1  Aggregate Task Component outputColumn Element

The Aggregate Task Component **outputColumn** element MUST NOT use any **name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **name** attribute values that are allowed for Aggregate Task Component on the **outputColumn** element, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| AggregationColumnId | An integer value that specifies the lineage identifier of a column that participates in a **Group By** or an aggregation operation. |
| AggregationType | An integer value that specifies the type of aggregation to be performed. The enumeration values are interpreted as follows:<br>0 – Group by<br>1 – Count<br>2 – Count all<br>3 – Count distinct<br>4 – Sum<br>5 – Average<br>6 – Minimum<br>7 – Maximum |
| IsBig | A Boolean value that specifies whether the column has a very large value (greater than 4 billion) or a very precise value (more precision than double precision).<br>TRUE specifies that the column contains a very large or very precise value.<br>FALSE specifies that the column does not contain a very large or very precise value. |
| AggregationComparisonFlags | An integer value that is a bitmask that specifies the comparison flags for the sort operation. The bit values are as follows:<br>0x00000001 - Ignore case.<br>0x00000002 - Ignore nonspace characters.<br>0x00001000 - Ignore Kana type.<br>0x00002000 - Ignore character width. |
| CountDistinctScale | An integer value that specifies the approximate number of groups that is expected by the operation. This value is used to optimize computation. The enumeration values are interpreted as follows:<br>0 – The key scale is not used.<br>1 – Aggregation writes 0 to 500,000 keys.<br>2 – Aggregation writes 500,000 to 25,000,000 keys.<br>3 – Aggregation writes more than 25,000,000 keys. |
| CountDistinctKeys | An integer value that specifies the exact number of **Group By** keys that can be written. If this value is specified, it takes precedence over the value of the **CountDistinctScale** attribute. |

#### 2.7.1.1.1.1.4.1.1.2.1.1.2  Audit Component outputColumn Element

The Audit Component **outputColumn** element MUST NOT use any **name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **name** attribute values that are allowed for Audit Component on the **outputColumn** element, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| LineageItemSelected | An integer value that specifies the audit item that is selected for output. The allowed values are as follows:<br>0 – Execution instance GUID.<br>1 – Package ID.<br>2 – Package name.<br>3 – Version ID.<br>4 - Execution start time.<br>5 - Machine name.<br>6 - User name.<br>7 - Task name.<br>8 - Task ID. |

#### 2.7.1.1.1.1.4.1.1.2.1.1.3    Cache Transform Component outputColumn Element

The Cache Transform Component MUST NOT have any properties on the **outputColumn** element.

#### 2.7.1.1.1.1.4.1.1.2.1.1.4    Character Map Component outputColumn Element

The Character Map Component MUST NOT have any properties on the **outputColumn** element.

#### 2.7.1.1.1.1.4.1.1.2.1.1.5    Conditional Split Component outputColumn Element

The Conditional Split Component MUST NOT have any properties on the **outputColumn** element.

#### 2.7.1.1.1.1.4.1.1.2.1.1.6    Copy Column Component outputColumn Element

The Copy Column Component **outputColumn** element MUST NOT use any **name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **name** attribute values that are allowed for Copy Column Component on the **outputColumn** element, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| copyColumnId | An integer value that specifies the lineage identifier of the column to copy. |

#### 2.7.1.1.1.1.4.1.1.2.1.1.7    Data Conversion Component outputColumn Element

The Data Conversion Component **outputColumn** element MUST NOT use any **name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **name** attribute values that are allowed for Data Conversion Component on the **outputColumn** element, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| SourceInputColumnLineageID | An integer value that specifies the lineage identifier of the source input column. |
| FastParse | A Boolean value that specifies whether to use the locale-neutral fast parse |

| Name attribute value | Description |
|---|---|
| | algorithm.<br>TRUE specifies to use the fast parse algorithm.<br>FALSE specifies not to use the fast parse algorithm. |

#### 2.7.1.1.1.1.4.1.1.2.1.1.8　Data Mining Query Component outputColumn Element

The Data Mining Query Component MUST NOT have any properties on the **outputColumn** element.

#### 2.7.1.1.1.1.4.1.1.2.1.1.9　Derived Column Component outputColumn Element

The Derived Column Component **outputColumn** element MUST NOT use any **name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **name** attribute values that are allowed for Derived Column Component on the **outputColumn** element, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| Expression | A string value that specifies the expression by which the column is derived. |
| FriendlyExpression | A Boolean value that specifies whether to use the locale-neutral fast parse algorithm.<br>TRUE specifies to use the fast parse algorithm.<br>FALSE specifies not to use the fast parse algorithm. |

#### 2.7.1.1.1.1.4.1.1.2.1.1.10　Export Column Component outputColumn Element

The Export Column Component MUST NOT have any properties on the **outputColumn** element.

#### 2.7.1.1.1.1.4.1.1.2.1.1.11　Fuzzy Grouping Component outputColumn Element

The Fuzzy Grouping Component **outputColumn** element MUST NOT use any **name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **name** attribute values that are allowed for Fuzzy Grouping Component on the **outputColumn** element, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| ColumnType | An integer value that specifies the type of the column. The allowed values are as follows:<br>0 – The type of the column is undefined.<br>1 – The column is a key input.<br>2 – The column is a key output.<br>3 – The column contains the similarity value.<br>4 – The column contains the column similarity value.<br>5 – The column is passthrough.<br>6 – The column is canonical. |
| InputId | An integer value that specifies the lineage identifier of the corresponding input column. |

#### 2.7.1.1.1.1.4.1.1.2.1.1.12　Fuzzy Lookup Component outputColumn Element

The Fuzzy Lookup Component **outputColumn** element MUST NOT use any **name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **name** attribute values that are allowed for Fuzzy Lookup Component on the **outputColumn** element, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| CopyFromReferenceColumn | A string value that specifies the name of the column in the reference table that provides the values for this output column. |
| SourceInputColumnLineageID | An integer value that specifies the lineage identifier of the source input column. |
| ColumnType | An integer value that specifies the type of the column.<br>0 – The type of the column is undefined.<br>1 – The column contains the similarity value.<br>2 – The column contains the confidence value.<br>3 – The column contains the column similarity value. |
| InputId | An integer value that specifies the lineage identifier of the corresponding input column. |

#### 2.7.1.1.1.1.4.1.1.2.1.1.13   Import Column Component outputColumn Element

The Import Column Component MUST NOT have any properties on the **outputColumn** element.

#### 2.7.1.1.1.1.4.1.1.2.1.1.14   Lookup Component outputColumn Element

The Lookup Component **outputColumn** element MUST NOT use any **name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **name** attribute values that are allowed for Lookup Component on the **outputColumn** element, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| CopyFromReferenceColumn | A string value that specifies the name of the column in the reference table that provides the values for this output column. |

#### 2.7.1.1.1.1.4.1.1.2.1.1.15   Merge Component outputColumn Element

The Merge Component **outputColumn** element MUST NOT use any **name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **name** attribute values that are allowed for Merge Component on the **outputColumn** element, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| OutputColumnLineageID | A string value that specifies the lineage identifier of the output column. |

#### 2.7.1.1.1.1.4.1.1.2.1.1.16   MergeJoin Component outputColumn Element

The MergeJoin Component **outputColumn** element MUST NOT use any **name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **name** attribute values that are allowed for MergeJoin Component on the **outputColumn** element, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| InputColumnID | A string value that specifies the lineage identifier of the input column from which data is copied to this output column. |

#### 2.7.1.1.1.1.4.1.1.2.1.1.17   Multicast Component outputColumn Element

The Multicast Component MUST NOT have any properties on the **outputColumn** element.

#### 2.7.1.1.1.1.4.1.1.2.1.1.18   OLE DB Command Component outputColumn Element

The OLE DB Command Component MUST NOT have any properties on the **outputColumn** element.

#### 2.7.1.1.1.1.4.1.1.2.1.1.19   Percentage Sampling Component outputColumn Element

The Percentage Sampling Component MUST NOT have any properties on the **outputColumn** element.

#### 2.7.1.1.1.1.4.1.1.2.1.1.20   Pivot Component outputColumn Element

The Pivot Component **outputColumn** element MUST NOT use any **name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **name** attribute values that are allowed for Pivot Component on the **outputColumn** element, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| SourceColumn | An integer value that specifies the lineage identifier of the input column that contains a pivoted value.<59> |
| PivotKeyValue | A string value that specifies one of the possible set of pivot keys that are identified in the input. |

#### 2.7.1.1.1.1.4.1.1.2.1.1.21   RowCount Component outputColumn Element

The RowCount Component MUST NOT have any properties on the **outputColumn** element.

#### 2.7.1.1.1.1.4.1.1.2.1.1.22   Row Sampling Component outputColumn Element

The Row Sampling Component **outputColumn** element MUST NOT use any **name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **name** attribute values that are allowed for the Row Sampling Component on the **outputColumn** element, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| InputColumnLineageID | An integer value that specifies the lineage identifier of the input column that is the source of this output column. |

#### 2.7.1.1.1.1.4.1.1.2.1.1.23   Script Component outputColumn Element

The Script Component MUST NOT have any properties on the **outputColumn** element.

#### 2.7.1.1.1.1.4.1.1.2.1.1.24   Slowly Changing Dimension Component outputColumn Element

The Slowly Changing Dimension Component MUST NOT have any properties on the **outputColumn** element.

#### 2.7.1.1.1.1.4.1.1.2.1.1.25   Sort Component outputColumn Element

The Sort Component **outputColumn** element MUST NOT use any **name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **name** attribute values that are allowed for Sort Component on the **outputColumn** element, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| SortColumnID | An integer value that specifies the lineage identifier of the sort column. |

#### 2.7.1.1.1.1.4.1.1.2.1.1.26   Term Extraction Component outputColumn Element

The Term Extraction Component MUST NOT have any properties on the **outputColumn** element.

#### 2.7.1.1.1.1.4.1.1.2.1.1.27   Term Lookup Component outputColumn Element

The Term Lookup Component **outputColumn** element MUST NOT use any **name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **name** attribute values that are allowed for Term Lookup Component on the **outputColumn** element, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| CustomLineageID | An integer value that specifies the lineage identifier of the corresponding input column. |

#### 2.7.1.1.1.1.4.1.1.2.1.1.28   Union All Component outputColumn Element

The Union All Component MUST NOT have any properties on the **outputColumn** element.

#### 2.7.1.1.1.1.4.1.1.2.1.1.29   Unpivot Component outputColumn Element

The Unpivot Component **outputColumn** element MUST NOT use any **name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **name** attribute values that are allowed for Unpivot Component on the **outputColumn** element, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| PivotKey | A Boolean value that specifies whether the **PivotKeyValue** property of input columns is written to this output column.<br><br>TRUE specifies that the **PivotKeyValue** property of input columns is written to this output column.<br><br>FALSE specifies that the **PivotKeyValue** property of input columns is not written to this output column. |

#### 2.7.1.1.1.1.4.1.1.2.1.1.30  ADO.NET Source Component outputColumn Element

The ADO.Net Source Component MUST NOT have any properties on the **outputColumn** element.

#### 2.7.1.1.1.1.4.1.1.2.1.1.31  Excel Source Component outputColumn Element

The Excel Source Component MUST NOT have any properties on the **outputColumn** element.

#### 2.7.1.1.1.1.4.1.1.2.1.1.32  Flat File Source Component outputColumn Element

The Flat File Source Component **outputColumn** element MUST NOT use any **name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **name** attribute values that are allowed for Flat File Source Component on the **outputColumn** element, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| FastParse | A Boolean value that specifies whether the component uses the faster locale-insensitive parsing method.<br><br>TRUE specifies that the faster locale-insensitive parsing method is used.<br><br>FALSE specifies that the faster locale-insensitive parsing method is not used. |
| UseBinaryFormat | A Boolean value that specifies whether data from the source is in binary format<br><br>TRUE specifies that the data is in binary format.<br><br>FALSE specifies that the data is not in binary format. |

#### 2.7.1.1.1.1.4.1.1.2.1.1.33  OLE DB Source Component outputColumn Element

The OLE DB Source Component MUST NOT have any properties on the **outputColumn** element.

#### 2.7.1.1.1.1.4.1.1.2.1.1.34  Raw File Source Component outputColumn Element

The Raw File Source Component MUST NOT have any properties on the **outputColumn** element.

#### 2.7.1.1.1.1.4.1.1.2.1.1.35  XML Source Component outputColumn Element

The XML Source Component MUST NOT have any properties on the **outputColumn** element.

#### 2.7.1.1.1.1.4.1.2    PipelineComponentOutputPropertiesType

The **PipelineComponentOutputPropertiesType** complex type specifies a collection of properties for an **output** element that is part of a Pipeline component.

The following is the XSD for the **PipelineComponentOutputPropertiesType** complex type.

```
<xs:complexType name="PipelineComponentOutputPropertiesType">
```

```
        <xs:sequence>
          <xs:element name="property" form="unqualified"
                      type="DTS:PipelineComponentOutputPropertyType"
                      minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
```

The following table specifies the elements of the **PipelineComponentOutputPropertiesType** complex type.

| Element | Type definition | Description |
|---------|-----------------|-------------|
| property | PipelineComponentOutputPropertyType | A complex type that specifies a single property within a collection of output properties for an **output** element of a Pipeline component. |

#### 2.7.1.1.1.1.4.1.2.1   PipelineComponentOutputPropertyType

The **PipelineComponentOutputPropertyType** complex type specifies the structure in which the value for a single property of an output of a component is held. All outputs do not use the same set of properties, so this general structure is designed to hold the variety of properties that an output can have.

The following is the XSD for the **PipelineComponentOutputPropertyType** complex type.

```
      <xs:complexType name="PipelineComponentOutputPropertyType"
                      mixed="true">
        <xs:sequence>
          <xs:element name="arrayElements"
                      type="DTS:PipelineComponentArrayElementsType"
                      form="unqualified" minOccurs="0"/>
        </xs:sequence>
        <xs:attributeGroup ref="DTS:PipelineComponentPropertyAttributeGroup"/>
        <xs:attribute name="name" use="required"
                      type="DTS:PipelineComponentOutputPropertyNameEnum"
                      form="unqualified"/>
      </xs:complexType>
```

The following table specifies the elements of the **PipelineComponentOutputPropertyType** complex type.

| Element | Type definition | Description |
|---------|-----------------|-------------|
| arrayElements | PipelineComponentArrayElementsType | A complex type that contains a collection of elements that form an array. The collection constitutes the value for the property. This value is used only if the **isArray** attribute value is TRUE. |

The following table specifies the attributes of the **PipelineComponentOutputPropertyType** complex type.

| Attribute | Description |
|-----------|-------------|
| name | An enumeration value that specifies the name of the property. |
| PipelineComponentPropertyAttributeGroup | An attribute group that contains attributes that are common to all **property** elements for **input** elements, **output** elements, **inputColumn** elements, and **outputColumn** elements. |

### 2.7.1.1.1.1.4.1.3    PipelineComponentOutputPropertyNameEnum

The **PipelineComponentOutputPropertyNameEnum** type contains the allowed values for the **name** attribute on the **output** element of a component of a Pipeline Task executable. The allowed values for the **property** element for each task are more restricted than the values that are specified formally by the XSD.

Each **name** attribute value MUST NOT be used for any **output** element except for the components that are specified in the subsections to this section. Some components allow no properties on the **output** element.

The additional restrictions that are specified in the subsections to this section MUST be followed for the **name** attribute on the **property** element children of the **output** element.

The following is the XSD for the **PipelineComponentOutputPropertyNameEnum** type.

```
<xs:simpleType name="PipelineComponentOutputPropertyNameEnum">
  <xs:restriction base="xs:string">
    <!--Transforms-->
    <!--Aggregate Task-->
    <xs:enumeration value="KeyScale"/>
    <xs:enumeration value="Keys"/>
    <!--Audit (none)-->
    <!--Cache Transform (none)-->
    <!--Character Map (none)-->
    <!--Conditional Split-->
    <xs:enumeration value="Expression"/>
    <xs:enumeration value="FriendlyExpression"/>
    <xs:enumeration value="EvaluationOrder"/>
    <xs:enumeration value="IsDefaultOut"/>
    <!--Copy Column (none)-->
    <!--Data Conversion (none)-->
    <!--Data Mining Query(none)-->
    <!--Derived Column Query (none)-->
    <!--Export Column Query (none)-->
    <!--Fuzzy Grouping(none)-->
    <!--Fuzzy Lookup(none)-->
    <!--Import Column (none)-->
    <!--Lookup(none)-->
    <!--Merge (none)-->
    <!--MergeJoin (none)-->
    <!--Multicast (none)-->
    <!--OLE DB Command(none)-->
    <!--Percentage Sampling-->
    <xs:enumeration value="Selected"/>
    <!--Pivot (none)-->
    <!--RowCount (none)-->
    <!--Row Sampling-->
    <xs:enumeration value="Selected"/>
    <!--Script Component(none)-->
    <!--Slowly Changing Dimension(none)-->
    <!--Sort(none)-->
    <!--Term Extraction(none)-->
    <!--Term Lookup(none)-->
    <!--UnionAll (none)-->
    <!--Unpivot (none)-->
    <!--Data Sources(none)-->
    <!--ADO.Net Source(none)-->
    <!--Excel Source(none)-->
    <!--Flat File Source(none)-->
    <!--OLEDB Source Task(none)-->
    <!--Raw File Source(none)-->
    <!--XML Source-->
    <xs:enumeration value="RowsetID"/>
  </xs:restriction>
</xs:simpleType>
```

### 2.7.1.1.1.1.4.1.3.1   Aggregate Task Component output Element

The Aggregate Task Component **output** element MUST NOT use any **name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **name** attribute values that are allowed for the Aggregate Component, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| KeyScale | An integer value that specifies the approximate number of groups expected by the operation. This value is used to optimize computation. The enumeration values are interpreted as follows:<br>0 – The key scale is not used.<br>1 – The aggregation writes approximately 500,000 keys.<br>2 – The aggregation writes approximately 25,000,000 keys.<br>3 – The aggregation writes more than 25,000,000 keys. |
| Keys | An integer value that specifies the exact number of **Group By** keys that can be written. If this value is specified, it takes precedence over the value of the **KeyScale** attribute. |

### 2.7.1.1.1.1.4.1.3.2   Audit Component output Element

The Audit Component **output** element MUST NOT have any properties.

### 2.7.1.1.1.1.4.1.3.3   Cache Transform Component output Element

The Cache Transform Component **output** element MUST NOT have any properties.

### 2.7.1.1.1.1.4.1.3.4   Character Map Component output Element

The Character Map Component **output** element MUST NOT have any properties.

### 2.7.1.1.1.1.4.1.3.5   Conditional Split Component output Element

The Conditional Split Component **output** element MUST NOT use any **name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **name** attribute values that are allowed for Conditional Split Component, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| Expression | A string value that specifies a conditional expression that is evaluated to determine the conditional split. |
| FriendlyExpression | A string value that specifies a conditional expression that is evaluated to determine the conditional split. The **FriendlyExpression** attribute uses column names in the expression syntax, while the **Expression** attribute uses only column lineage identifiers. |
| EvaluationOrder | An integer value that specifies the order in which the expression is evaluated, from lowest to highest. |
| IsDefaultOut | A Boolean value that specifies whether an output is the default output.<br>TRUE specifies that the output is the default output.<br>FALSE specifies that the output is not the default output. |

### 2.7.1.1.1.1.4.1.3.6   Copy Column Component output Element

The Copy Column Component **output** element MUST NOT have any properties.

#### 2.7.1.1.1.1.4.1.3.7 Data Conversion Component output Element

The Data Conversion Component **output** element MUST NOT have any properties.

#### 2.7.1.1.1.1.4.1.3.8 Data Mining Query Component output Element

The Data Mining Query Component **output** element MUST NOT have any properties.

#### 2.7.1.1.1.1.4.1.3.9 Derived Column Component output Element

The Derived Column Component **output** element MUST NOT have any properties.

#### 2.7.1.1.1.1.4.1.3.10 Export Column Component output Element

The Export Column Component **output** element MUST NOT have any properties.

#### 2.7.1.1.1.1.4.1.3.11 Fuzzy Grouping Component output Element

The Fuzzy Grouping Component **output** element MUST NOT have any properties.

#### 2.7.1.1.1.1.4.1.3.12 Fuzzy Lookup Component output Element

The Fuzzy Lookup Component **output** element MUST NOT have any properties.

#### 2.7.1.1.1.1.4.1.3.13 Import Column Component output Element

The Import Column Component **output** element MUST NOT have any properties.

#### 2.7.1.1.1.1.4.1.3.14 Lookup Component output Element

The Lookup Component **output** element MUST NOT have any properties.

#### 2.7.1.1.1.1.4.1.3.15 Merge Component output Element

The Merge Component **output** element MUST NOT have any properties.

#### 2.7.1.1.1.1.4.1.3.16 MergeJoin Component output Element

The MergeJoin Component **output** element MUST NOT have any properties.

#### 2.7.1.1.1.1.4.1.3.17 Multicast Component output Element

The Multicast Component **output** element MUST NOT have any properties.

#### 2.7.1.1.1.1.4.1.3.18 OLE DB Command Component output Element

The OLE DB Command Component **output** element MUST NOT have any properties.

#### 2.7.1.1.1.1.4.1.3.19 Percentage Sampling Component output Element

The Percentage Sampling Component **output** element MUST NOT use any **name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **name** attribute values that are allowed for Percentage Sampling Component, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| Selected | A Boolean value that specifies to which output sampled output is directed. TRUE specifies that sampled output is directed to this output. |

| Name attribute value | Description |
|---|---|
| | FALSE specifies that sampled output is not directed to this output. |

### 2.7.1.1.1.1.4.1.3.20   Pivot Component output Element

The Pivot Component **output** element MUST NOT have any properties.

### 2.7.1.1.1.1.4.1.3.21   RowCount Component output Element

The RowCount Component **output** element MUST NOT have any properties.

### 2.7.1.1.1.1.4.1.3.22   Row Sampling Component output Element

The Row Sampling Component **output** element MUST NOT use any **name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **name** attribute values that are allowed for Row Sampling Component, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| Selected | A Boolean value that specifies to which output sampled output is directed. TRUE specifies that sampled output is directed to this output. FALSE specifies that sampled output is not directed to this output. |

### 2.7.1.1.1.1.4.1.3.23   Script Component output Element

The Script Component **output** element MUST NOT have any properties.

### 2.7.1.1.1.1.4.1.3.24   Slowly Changing Dimension Component output Element

The Slowly Changing Dimension Component **output** element MUST NOT have any properties.

### 2.7.1.1.1.1.4.1.3.25   Sort Component output Element

The Sort Component **output** element MUST NOT have any properties.

### 2.7.1.1.1.1.4.1.3.26   Term Extraction Component output Element

The Term Extraction Component **output** element MUST NOT have any properties.

### 2.7.1.1.1.1.4.1.3.27   Term Lookup Component output Element

The Term Lookup Component **output** element MUST NOT have any properties.

### 2.7.1.1.1.1.4.1.3.28   Union All Component output Element

The Union All Component **output** element MUST NOT have any properties.

### 2.7.1.1.1.1.4.1.3.29   Unpivot Component output Element

The Unpivot Component **output** element MUST NOT have any properties.

### 2.7.1.1.1.1.4.1.3.30   ADO.NET Source Component output Element

The ADO.Net Source Component **output** element MUST NOT have any properties.

### 2.7.1.1.1.1.4.1.3.31   Excel Source Component output Element

The Excel Source Component **output** element MUST NOT have any properties.

### 2.7.1.1.1.4.1.3.32 Flat File Source Component output Element

The Flat File Source Component **output** element MUST NOT have any properties.

### 2.7.1.1.1.4.1.3.33 OLE DB Source Component output Element

The OLE DB Source Component **output** element MUST NOT have any properties.

### 2.7.1.1.1.4.1.3.34 Raw File Source Component output Element

The Raw File Source Component **output** element MUST NOT have any properties.

### 2.7.1.1.1.4.1.3.35 XML Source Component output Element

The XML Source Component **output** element MUST NOT use any **name** attribute value on a **property** element except for those that are listed in the table in this section.

The following table specifies the **name** attribute values that are allowed for XML Source Component, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| RowsetID | A string value that specifies the name of the rowset that is associated with the output. |

## 2.7.1.1.1.1.5 PipelineComponentComponentClassIDEnum

The **PipelineComponentComponentClassIDEnum** type contains the enumeration values for the **componentClassID** attribute.

The following is the XSD for the **PipelineComponentComponentClassIDEnum** type.

```
<xs:simpleType name="PipelineComponentComponentClassIDEnum">
  <xs:restriction base="DTS:uuid">
    <!--Transforms-->
    <!--Aggregate-->
    <xs:enumeration value="{5B201335-B360-485C-BB93-75C34E09B3D3}"/>
    <!--Audit-->
    <xs:enumeration value="{4FCE8297-9C65-40A9-9D60-00C5CC6BC283}"/>
    <!--Cache Transform-->
    <xs:enumeration value="{BF818E79-2C1C-410D-ADEA-B2D1A04FED01}"/>
    <!--Character Map-->
    <xs:enumeration value="{C86C867A-7D40-486A-8734-F0B6943A9DD4}"/>
    <!--Conditional Split-->
    <xs:enumeration value="{7F88F654-4E20-4D14-84F4-AF9C925D3087}"/>
    <!--Copy Column-->
    <xs:enumeration value="{D231ABC4-4754-481B-8F7A-CD4BD8D09832}"/>
    <!--Data Conversion-->
    <xs:enumeration value="{62B1106C-7DB8-4EC8-ADD6-4C664DFFC54A}"/>
    <!--Data Mining Query-->
    <xs:enumeration value="{CBDFF0CC-3758-4424-A431-3C2E103069AA}"/>
    <!--Derived Column-->
    <xs:enumeration value="{49928E82-9C4E-49F0-AABE-3812B82707EC}"/>
    <!--Export Column-->
    <xs:enumeration value="{A9823821-0085-4483-952F-36277F63FFC5}"/>
    <!--Fuzzy Grouping-->
    <xs:enumeration value="{836E0D0B-0358-45B4-B888-4ADCE9356D4E}"/>
    <!--Fuzzy Lookup-->
    <xs:enumeration value="{E4A5F949-EC93-45AB-8B36-B52936257EC2}"/>
    <!--Import Column-->
    <xs:enumeration value="{9053CAE0-140B-48E4-AEEC-4E6A26E9E436}"/>
    <!--Lookup-->
    <xs:enumeration value="{671046B0-AA63-4C9F-90E4-C06E0B710CE3}"/>
```

```
        <!--Merge-->
        <xs:enumeration value="{36E0E750-2510-4776-AA6E-17EAE84FD63E}"/>
        <!--MergeJoin-->
        <xs:enumeration value="{14D43A4F-D7BD-489D-829E-6DE35750CFE4}"/>
        <!--Multicast-->
        <xs:enumeration value="{EC139FBC-694E-490B-8EA7-35690FB0F445}"/>
        <!--OLE DB Command-->
        <xs:enumeration value="{93FFEC66-CBC8-4C7F-9C6A-CB1C17A7567D}"/>
        <!--Percentage Sampling-->
        <xs:enumeration value="{05322586-D434-4593-8BE4-8CFC766D9C8B}"/>
        <!--Pivot-->
        <xs:enumeration value="{AEB2AFAE-2201-48B1-B290-42BB8C9149DD}"/>
        <!--RowCount-->
        <xs:enumeration value="{E2697D8C-70DA-42B2-8208-A19CE3A9FE41}"/>
        <!--Row Sampling-->
        <xs:enumeration value="{284A868A-7C3D-498C-ABBF-028CEF6AB288}"/>
        <!--Script Component, ADO.Net source, XML Source,ADO.Net
        Destination, DataReader Destination, SQL Server Compact Destination-->
        <xs:enumeration value="{874F7595-FB5F-40FF-96AF-FBFF8250E3EF}"/>
        <!--Slowly Changing Dimension-->
        <xs:enumeration value="{25BBB0C5-369B-4303-B3DF-D0DC741DEE58}"/>
        <!--Sort-->
        <xs:enumeration value="{5B1A3FF5-D366-4D75-AD1F-F19A36FCBEDB}"/>
        <!--Term Extraction-->
        <xs:enumeration value="{056F53D1-8081-4D07-8134-1D4DAC244BB2}"/>
        <!--Term Lookup-->
        <xs:enumeration value="{ACCC729A-823D-49A7-B16D-F1999AEB1D4C}"/>
        <!--Union All-->
        <xs:enumeration value="{B594E9A8-4351-4939-891C-CFE1AB93E925}"/>
        <!--Unpivot-->
        <xs:enumeration value="{B8B094A9-8809-4E06-8874-5C55606A9FDF}"/>
        <!--Destinations-->
        <!--Excel Destination-->
        <xs:enumeration value="{1F5D5712-2FBA-4CB9-A95A-86C1F336E1DA}"/>
        <!--Flat File Destination-->
        <xs:enumeration value="{8DA75FED-1B7C-407D-B2AD-2B24209CCCA4}"/>
        <!--OLE DB Destination-->
        <xs:enumeration value="{4ADA7EAA-136C-4215-8098-D7A7C27FC0D1}"/>
        <!--Data Mining Model Training-->
        <xs:enumeration value="{3D9FFAE9-B89B-43D9-80C8-B97D2740C746}"/>
        <!--Dimension Processing-->
        <xs:enumeration value="{2C2F0891-3AAA-4865-A676-D7476FE4CE90}"/>
        <!--Partition Processing-->
        <xs:enumeration value="{DA510FB7-E3A8-4D96-9F59-55E15E67FE3D}"/>
        <!--Raw File Destination-->
        <xs:enumeration value="{04762BB6-892F-4EE6-AD46-9CEB0A7EC7A2}"/>
        <!--RecordSet Destination-->
        <xs:enumeration value="{C457FD7E-CE98-4C4B-AEFE-F3AE0044F181}"/>
        <!--SQL Server Destination-->
        <xs:enumeration value="{F452EAF3-5EF0-43F1-8067-09DDF0BC6316}"/>
        <!--Sources-->
        <!--Excel Source-->
        <xs:enumeration value="{8C084929-27D1-479F-9641-ABB7CDADF1AC}"/>
        <!--Flat File Source-->
        <xs:enumeration value="{D23FD76B-F51D-420F-BBCB-19CBF6AC1AB4}"/>
        <!--OLE DB Source-->
        <xs:enumeration value="{165A526D-D5DE-47FF-96A6-F8274C19826B}"/>
        <!--Raw File Source-->
        <xs:enumeration value="{480C7D5A-CE63-405C-B338-3C7F26560EE3}"/>
    </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the
**PipelineComponentComponentClassIDEnum** type.

| Enumeration value | Description |
|---|---|
| {5B201335-B360-485C-BB93-75C34E09B3D3} | Aggregate Component |
| {4FCE8297-9C65-40A9-9D60-00C5CC6BC283} | Audit Component |
| {BF818E79-2C1C-410D-ADEA-B2D1A04FED01} | Cache Transform Component |
| {C86C867A-7D40-486A-8734-F0B6943A9DD4} | Character Map Component |
| {7F88F654-4E20-4D14-84F4-AF9C925D3087} | Conditional Split Component |
| {D231ABC4-4754-481B-8F7A-CD4BD8D09832} | Copy Column Component |
| {62B1106C-7DB8-4EC8-ADD6-4C664DFFC54A} | Data Conversion Component |
| {CBDFF0CC-3758-4424-A431-3C2E103069AA} | Data Mining Query Component |
| {49928E82-9C4E-49F0-AABE-3812B82707EC} | Derived Column Component |
| {A9823821-0085-4483-952F-36277F63FFC5} | Export Column Component |
| {836E0D0B-0358-45B4-B888-4ADCE9356D4E} | Fuzzy Grouping Component |
| {E4A5F949-EC93-45AB-8B36-B52936257EC2} | Fuzzy Lookup Component |
| {9053CAE0-140B-48E4-AEEC-4E6A26E9E436} | Import Column Component |
| {671046B0-AA63-4C9F-90E4-C06E0B710CE3} | Lookup Component |
| {36E0E750-2510-4776-AA6E-17EAE84FD63E} | Merge Component |
| {14D43A4F-D7BD-489D-829E-6DE35750CFE4} | MergeJoin Component |
| {EC139FBC-694E-490B-8EA7-35690FB0F445} | Multicast Component |
| {93FFEC66-CBC8-4C7F-9C6A-CB1C17A7567D} | OLE DB Command Component |
| {05322586-D434-4593-8BE4-8CFC766D9C8B} | Percentage Sampling Component |
| {AEB2AFAE-2201-48B1-B290-42BB8C9149DD} | Pivot Component |
| {E2697D8C-70DA-42B2-8208-A19CE3A9FE41} | RowCount Component |
| {284A868A-7C3D-498C-ABBF-028CEF6AB288} | Row Sampling Component |
| {874F7595-FB5F-40FF-96AF-FBFF8250E3EF}<60> | Script Component<br>ADO.Net Source Component<br>XML Source Component<br>ADO.Net Destination Component<br>DataReader Destination Component<br>SQL Server Compact Destination Component |
| {25BBB0C5-369B-4303-B3DF-D0DC741DEE58} | Slowly Changing Dimension Component |
| {5B1A3FF5-D366-4D75-AD1F-F19A36FCBEDB} | Sort Component |
| {056F53D1-8081-4D07-8134-1D4DAC244BB2} | Term Extraction Component |
| {ACCC729A-823D-49A7-B16D-F1999AEB1D4C} | Term Lookup Component |
| {B594E9A8-4351-4939-891C-CFE1AB93E925} | Union All Component |

| Enumeration value | Description |
|---|---|
| {B8B094A9-8809-4E06-8874-5C55606A9FDF} | Unpivot Component |
| {1F5D5712-2FBA-4CB9-A95A-86C1F336E1DA} | Excel Destination Component |
| {8DA75FED-1B7C-407D-B2AD-2B24209CCCA4} | Flat File Destination Component |
| {4ADA7EAA-136C-4215-8098-D7A7C27FC0D1} | OLE DB Destination Component |
| {3D9FFAE9-B89B-43D9-80C8-B97D2740C746} | Data Mining Model Training Destination Component |
| {2C2F0891-3AAA-4865-A676-D7476FE4CE90} | Dimension Processing Destination Component |
| {DA510FB7-E3A8-4D96-9F59-55E15E67FE3D} | Partition Processing Destination Component |
| {04762BB6-892F-4EE6-AD46-9CEB0A7EC7A2} | Raw File Destination Component |
| {C457FD7E-CE98-4C4B-AEFE-F3AE0044F181} | RecordSet Destination Component |
| {F452EAF3-5EF0-43F1-8067-09DDF0BC6316} | SQL Server Destination Component |
| {8C084929-27D1-479F-9641-ABB7CDADF1AC} | Excel Source Component |
| {D23FD76B-F51D-420F-BBCB-19CBF6AC1AB4} | Flat File Source Component |
| {165A526D-D5DE-47FF-96A6-F8274C19826B} | OLE DB Source Component |
| {480C7D5A-CE63-405C-B338-3C7F26560EE3} | Raw File Source Component |

### 2.7.1.1.1.1.6 Attribute Groups and Common Types Used by Components

The following sections specify attribute groups and other types that are referenced by type definitions within the Pipeline component's hierarchy of types.

#### 2.7.1.1.1.1.6.1 PipelineComponentMetadataColumnsType

The **PipelineComponentMetadataColumnsType** complex type specifies a collection of external metadata column objects for an **input** element or an **output** element of a Pipeline Task executable.

The following is the XSD for the **PipelineComponentMetadataColumnsType** complex type.

```
<xs:complexType name="PipelineComponentMetadataColumnsType">
  <xs:sequence>
    <xs:element name="externalMetadataColumn"
        type="DTS:PipelineComponentMetadataColumnType"
        minOccurs="0" maxOccurs="unbounded" form="unqualified"/>
  </xs:sequence>
  <xs:attribute name="isUsed" form="unqualified">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="True"/>
        <xs:enumeration value="False"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:complexType>
```

The following table specifies the elements of the **PipelineComponentMetadataColumnsType** complex type.

| Element | Type definition | Description |
|---|---|---|
| externalMetadataColumn | PipelineComponentMetadataColumnType | A complex type that specifies a single metadata column definition within an input or an output of a Pipeline. |

The following table specifies the attributes of the **PipelineComponentMetadataColumnsType** complex type.

| Attribute group | Description |
|---|---|
| isUsed | An enumeration value that specifies whether external metadata columns are used. "True" specifies that external metadata columns are used. "False" specifies that external metadata columns are not used. |

### 2.7.1.1.1.1.6.1.1 PipelineComponentMetadataColumnType

The **PipelineComponentMetadataColumnType** complex type specifies a single column that is a member of a collection of metadata columns that are part of a Pipeline component.

The following is the XSD for the PipelineComponentMetadataColumnType complex type.

```
    <xs:complexType name="PipelineComponentMetadataColumnType">
      <xs:sequence>
        <xs:element name="properties"
type="DTS:PipelineComponentMetadataColumnPropertiesType" minOccurs="0"
form="unqualified"/>
      </xs:sequence>
      <xs:attributeGroup ref="DTS:PipelineComponentAllColumnBaseAttributeGroup"/>
      <xs:attributeGroup ref="DTS:PipelineComponentColumnExtendedAttributeGroup"/>
    </xs:complexType>
```

The following table specifies the elements of the **PipelineComponentMetadataColumnType** complex type.

| Element | Type definition | Description |
|---|---|---|
| properties | PipelineComponentMetadataColumnPropertiesType | A complex type that specifies a collection of properties of a metadata column. |

The following table specifies the attribute groups of the **PipelineComponentMetadataColumnType** type.

| Attribute group | Description |
|---|---|
| PipelineComponentAllColumnBaseAttributeGroup | An attribute group specification that specifies attributes that are common to all types of columns for Pipeline components. |
| PipelineComponentColumnExtendedAttributeGroup | An attribute group specification that specifies additional attributes that are present for the **PipelineComponentMetadataColumnType** type. |

### 2.7.1.1.1.1.6.1.1.1 PipelineComponentMetadataColumnPropertiesType

The **PipelineComponentMetadataColumnPropertiesType** complex type specifies a collection of properties for an **externalMetadataColumn** element that is part of a Pipeline component.

The following is the XSD for the **PipelineComponentMetadataColumnPropertiesType** complex type.

```
<xs:complexType name="PipelineComponentMetadataColumnPropertiesType">
  <xs:sequence>
    <xs:element name="property"
        type="DTS:PipelineComponentMetadataColumnPropertyType"
        minOccurs="0" maxOccurs="unbounded" form="unqualified"/>
  </xs:sequence>
</xs:complexType>
```

The following table specifies the elements of the **PipelineComponentMetadataColumnPropertiesType** complex type.

| Element | Type definition | Description |
|---|---|---|
| property | PipelineComponentMetadataColumnPropertyType | A complex type that specifies a single property within a collection of metadata column properties for an **externalMetadataColumn** element of a Pipeline component. |

#### 2.7.1.1.1.1.6.1.1.1.1 PipelineComponentMetadataColumnPropertyType

The **PipelineComponentMetadataColumnPropertyType** complex type specifies a property of an inputColumn element.

The following is the XSD for the **PipelineComponentMetadataColumnPropertyType** complex type.

```
<xs:complexType name="PipelineComponentMetadataColumnPropertyType">
  <xs:simpleContent>
    <xs:extension base="xs:anySimpleType">
      <xs:attributeGroup ref="DTS:PipelineComponentPropertyAttributeGroup"/>
      <xs:attribute name="name" form="unqualified" use="required"
          type="DTS:PipelineComponentMetadataColumnPropertyNameEnum"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

The following table specifies the attributes and attribute groups of the **PipelineComponentMetadataColumnPropertyType** complex type.

| Attribute | Description |
|---|---|
| PipelineComponentPropertyAttributeGroup | An attribute group that contains attributes that are common to all property elements for **input** elements, **output** elements, **inputColumn** elements, and **outputColumn** elements. |
| name | A string value that specifies the name of the property. |

#### 2.7.1.1.1.1.6.1.1.1.1.1 PipelineComponentMetadataColumnPropertyNameEnum

The **PipelineComponentMetadataColumnPropertyNameEnum** type contains the allowed values for the **name** attribute on the **externalMetadataColumn** element of a Pipeline component. The allowed values for the **property** element for each task are more restricted than the values that are specified formally by the XSD.

Each **name** attribute value MUST NOT be used for any component except for the components that are specified in the subsections to this section. Some components allow no properties.

The additional restrictions specified in the subsections to this section MUST be followed for the **name** attribute on the **property** element children of the **externalMetadataColumn** element.

The following is the XSD of the **PipelineComponentMetadataColumnPropertyNameEnum** type.

```
<xs:simpleType name="PipelineComponentMetadataColumnPropertyNameEnum">
  <xs:restriction base="xs:string">
    <!--OLE DB Command-->
    <xs:enumeration value="DBParamInfoFlags"/>
    <!--XML Source-->
    <xs:enumeration value="CLRType"/>
  </xs:restriction>
</xs:simpleType>
```

#### 2.7.1.1.1.1.6.1.1.1.1.1.2   OLE DB Command Component

The OLE DB Command Component **externalMetadataColumn** element MUST NOT use any **name** attribute value on a child **property** element except for those that are listed in the table in this section.

The following table specifies the **name** attribute values that are allowed for OLE DB Command Component on the **externalMetadataColumn** element, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| DBParamInfoFlags | An integer value that is a bitmask that specifies characteristics of a parameter. The interpretation of the bits in the bitmask is as follows: 0x00000001 – The parameter accepts values on input. 0x00000002 – The parameter returns values on output. 0x00000010 – The parameter is signed. 0x00000040 – The parameter accepts NULL values. 0x00000080 – The parameter is a binary large object (BLOB). 0x00000100 – The parameter value represents the absolute value of the negative scale of the parameter. |

#### 2.7.1.1.1.1.6.1.1.1.1.1.3   XML Source Component

The XML Source Component **externalMetadataColumn** element MUST NOT use any **name** attribute value on a child **property** element except for those that are listed in the table in this section.

The following table specifies the **name** attribute values that are allowed for XML Source Component on the **externalMetadataColumn** element, in addition to any restrictions on the element value. These restrictions MUST be followed.

| Name attribute value | Description |
|---|---|
| CLRType | A string value that specifies the type of the column, in terms of the CLR type system. The allowed string values are the following:  ▪  System.Boolean – Boolean  ▪  System.Byte – Byte |

| Name attribute value | Description |
|---|---|
| | ▪ System.SByte – 8-bit signed integer |
| | ▪ System.Char – Unicode character |
| | ▪ System.Decimal – Decimal number |
| | ▪ System.Double – Double-precision real number |
| | ▪ System.Single – Single-precision real number |
| | ▪ System.Int32 – 32-bit integer |
| | ▪ System.UInt32 – Unsigned 32-bit integer |
| | ▪ System.Int64 – 64-bit integer |
| | ▪ System.UInt64 – Unsigned 64-bit integer |
| | ▪ System.Object – Object |
| | ▪ System.Int16 - 16-bit integer |
| | ▪ System.UInt16 – Unsigned 16-bit integer |
| | ▪ System.String – Unicode character string |
| | For more information about Microsoft .NET Framework system types, see [MSDN-TYPCLS]. |

### 2.7.1.1.1.1.6.2    PipelineComponentRowDispositionEnum

The **PipelineComponentRowDispositionEnum** type contains the enumeration values for the **state** attribute.

The following is the XSD for the **PipelineComponentRowDispositionEnum** type.

```
<xs:simpleType name="PipelineComponentRowDispositionEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="NotUsed"/>
    <xs:enumeration value="IgnoreFailure"/>
    <xs:enumeration value="FailComponent"/>
    <xs:enumeration value="RedirectRow"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **PipelineComponentRowDispositionEnum** type.

| Enumeration value | Description |
|---|---|
| NotUsed | Specifies that row processing is not used. |
| IgnoreFailure | Specifies that the component ignores the failure and continue processing. |
| FailComponent | Specifies that the component's execution fails. |
| RedirectRow | Specifies that any row that causes a failure is redirected to the error output. |

#### 2.7.1.1.1.1.6.3 PipelineComponentInputOutputElementAttributeGroup

The **PipelineComponentInputOutputElementAttributeGroup** attribute group specifies attributes that are used in common by the **input** element and the **output** element of the Pipeline component.

The following is the XSD for the **PipelineComponentInputOutputElementAttributeGroup** attribute group.

```
   <xs:attributeGroup name="PipelineComponentInputOutputElementAttributeGroup">
     <xs:attribute name="refId" type="xs:string" use="required" form="unqualified"/>
     <xs:attribute name="name" type="xs:string" use="optional" default=""
form="unqualified"/>
     <xs:attribute name="description" type="xs:string" use="optional" default=""
form="unqualified"/>
     <xs:attribute name="hasSideEffects" type="xs:boolean" use="optional" default="false"
form="unqualified"/>
     <xs:attribute name="dangling" type="xs:boolean" use="optional" default="false"
form="unqualified"/>
     <xs:attribute name="errorOrTruncationOperation" type="xs:string" use="optional"
default="" form="unqualified"/>
     <xs:attribute name="errorRowDisposition" form="unqualified"
type="DTS:PipelineComponentRowDispositionEnum" use="optional" default="NotUsed"/>
     <xs:attribute name="truncationRowDisposition" form="unqualified"
type="DTS:PipelineComponentRowDispositionEnum" use="optional" default="NotUsed"/>
   </xs:attributeGroup>
```

The following table specifies additional information for the attributes of the **PipelineComponentInputOutputElementAttributeGroup** attribute group.

| Attribute | Description |
|---|---|
| refId | A string value that specifies a unique **refId** for this object. Elsewhere within the containing executable or within the package, the value of **refId** can be used to refer to this object.[61] |
| name | A string value that specifies the name for this column element. |
| description | A string value that specifies the description for this component **output** element. |
| hasSideEffects | A Boolean value that indicates whether a component can be removed from the execution plan of the data flow when it is not attached to an upstream component and when the value of the **runInOptimizedMode** attribute is TRUE.<br><br>TRUE indicates that the component cannot be removed from the execution plan when it is not attached to an upstream component and when the **runInOptimizedMode** attribute of the **Pipeline** element is TRUE.<br><br>FALSE indicates that the component can be removed from the execution plan when it is not attached to an upstream component and when the **runInOptimizedMode** attribute of the **Pipeline** element is TRUE. |

| Attribute | Description |
|---|---|
| dangling | A Boolean value that specifies whether the output is dangling, that is, that it does not lead to another component.<br>TRUE specifies that the output is dangling.<br>FALSE specifies that the output is not dangling. |
| errorOrTruncationOperation | A string value that specifies a string that can be displayed by the user interface to indicate the type of operation that causes an error or truncation operation to occur. |
| errorRowDisposition | An enumeration value that specifies the handling of row errors. |
| truncationRowDisposition | An enumeration value that specifies the handling of truncations that occur when processing rows. |

#### 2.7.1.1.1.1.6.4 PipelineComponentAllColumnBaseAttributeGroup

The **PipelineComponentAllColumnBaseAttributeGroup** attribute group specifies the attributes that are used for all columns that are specified by components.

The following is the XSD for the **PipelineComponentAllColumnBaseAttributeGroup** attribute group.

```
<xs:attributeGroup name="PipelineComponentAllColumnBaseAttributeGroup">
  <xs:attribute name="refId" type="xs:string" use="required" form="unqualified"/>
  <xs:attribute name="name" type="xs:string" use="optional" default=""
form="unqualified"/>
  <xs:attribute name="description" type="xs:string" use="optional" default=""
form="unqualified"/>
  <xs:attribute name="mappedColumnId" type="xs:int" use="optional" default="0"
form="unqualified"/>
</xs:attributeGroup>
```

The following table specifies additional information for the attributes of the **PipelineComponentAllColumnBaseAttributeGroup** attribute group.

| Attribute | Description |
|---|---|
| refId | A string value that specifies a unique **refId** for this object. Elsewhere within the containing executable or within the package, the value of **refId** can be used to refer to this object.<62> |
| name | A string value that specifies the name for this **column** element. |
| description | A string value that specifies the description for this **column** element. |
| mappedColumnId | An integer value that specifies the identifier of a column that is mapped to this column. |

#### 2.7.1.1.1.1.6.5 PipelineComponentIOColumnBaseAttributeGroup

The **PipelineComponentIOColumnBaseAttributeGroup** attribute group specifies the attributes that are used for all columns that serve as input columns or output columns that are specified by components.

The following is the XSD for the **PipelineComponentIOColumnBaseAttributeGroup** attribute group.

```
<xs:attributeGroup name="PipelineComponentIOColumnBaseAttributeGroup">
  <xs:attribute name="errorOrTruncationOperation" type="xs:string"
                use="optional" default="" form="unqualified"/>
  <xs:attribute name="errorRowDisposition"
                type="DTS:PipelineComponentRowDispositionEnum"
                form="unqualified" use="optional" default="NotUsed"/>
  <xs:attribute name="truncationRowDisposition"
                type="DTS:PipelineComponentRowDispositionEnum"
                form="unqualified" use="optional" default="NotUsed"/>
  <xs:attribute name="externalMetadataColumnId" type="xs:string" form="unqualified"/>
  <xs:attribute name="lineageId" type="xs:string" use="required" form="unqualified"/>
</xs:attributeGroup>
```

The following table specifies additional information for the attributes of the
**PipelineComponentIOColumnBaseAttributeGroup** attribute group.

| Attribute | Description |
|---|---|
| errorOrTruncationOperation | A string value that specifies a string that can be displayed by the user interface to indicate the type of operation that causes an error or truncation operation to occur. |
| errorRowDisposition | An enumeration value that specifies the handling of row errors. |
| truncationRowDisposition | An enumeration value that specifies the handling of truncations that occur when processing rows. |
| externalMetadataColumnId | A string value that specifies the identifier of the external metadata column for this column. |
| lineageId | A string value.<63> |

#### 2.7.1.1.1.1.6.6　　　　PipelineComponentColumnExtendedAttributeGroup

The **PipelineComponentColumnExtendedAttributeGroup** attribute group specifies additional attributes that are used for some columns that are specified within components.

The following is the XSD for the **PipelineComponentColumnExtendedAttributeGroup** attribute group.

```
<xs:attributeGroup name="PipelineComponentColumnExtendedAttributeGroup">
  <xs:attribute name="precision" type="xs:int" use="optional" default="0"
                form="unqualified"/>
  <xs:attribute name="scale" type="xs:int" use="optional" default="0"
                form="unqualified"/>
  <xs:attribute name="length" type="xs:int" use="optional" default="0"
                form="unqualified"/>
  <xs:attribute name="dataType"
                type="DTS:PipelineComponentColumnDataTypeEnum"
                use="optional" default="empty" form="unqualified"/>
  <xs:attribute name="codePage" type="xs:int" use="optional" default="0"
                form="unqualified"/>
</xs:attributeGroup>
```

The following table specifies additional information for the attributes of the
**PipelineComponentColumnExtendedAttributeGroup** attribute group.

| Attribute | Description |
|-----------|-------------|
| precision | An integer value that specifies the total digits in a numeric data type. |
| scale | An integer value that specifies the number of digits to the right of the decimal point. |
| length | An integer value that specifies the length of the column in characters. |
| dataType | An enumeration value that specifies the type of the data in the column. |
| codePage | An integer value that specifies the numeric value of the ANSI code page that is used for strings for this column. |

#### 2.7.1.1.1.1.6.6.1 PipelineComponentColumnDataTypeEnum

The **PipelineComponentColumnDataTypeEnum** type contains the enumeration values for the **DataType** attribute as it appears within the different types of column elements.

The following is the XSD for the **PipelineComponentColumnDataTypeEnum** type.

```
<xs:simpleType name="PipelineComponentColumnDataTypeEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="empty"/>
    <xs:enumeration value="i1"/>
    <xs:enumeration value="ui1"/>
    <xs:enumeration value="i2"/>
    <xs:enumeration value="i4"/>
    <xs:enumeration value="r4"/>
    <xs:enumeration value="r8"/>
    <xs:enumeration value="cy"/>
    <xs:enumeration value="date"/>
    <xs:enumeration value="bool"/>
    <xs:enumeration value="variant"/>
    <xs:enumeration value="decimal"/>
    <xs:enumeration value="ui2"/>
    <xs:enumeration value="ui4"/>
    <xs:enumeration value="i8"/>
    <xs:enumeration value="ui8"/>
    <xs:enumeration value="guid"/>
    <xs:enumeration value="bytes"/>
    <xs:enumeration value="str"/>
    <xs:enumeration value="wstr"/>
    <xs:enumeration value="numeric"/>
    <xs:enumeration value="dbDate"/>
    <xs:enumeration value="dateTime"/>
    <xs:enumeration value="dbTime"/>
    <xs:enumeration value="dbTimeStamp"/>
    <xs:enumeration value="image"/>
    <xs:enumeration value="text"/>
    <xs:enumeration value="nText"/>
    <xs:enumeration value="dbTime2"/>
    <xs:enumeration value="dbTimeStamp2"/>
    <xs:enumeration value="dbTimeStampOffset"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **PipelineComponentColumnDataTypeEnum** type.

| Enumeration value | Description |
| --- | --- |
| empty | Empty value. |
| i1 | One-byte integer. |
| ui1 | One-byte unsigned integer. |
| i2 | Two-byte integer. |
| i4 | Four-byte integer. |
| r4 | Four-byte real number. |
| r8 | Eight-byte real number. |
| cy | Calendar year. |
| date | Date. |
| bool | Boolean. |
| variant | Variant. |
| decimal | Decimal value. |
| ui2 | Two-byte unsigned integer. |
| ui4 | Four-byte unsigned integer. |
| i8 | Eight-byte integer. |
| ui8 | Eight-byte unsigned integer. |
| guid | GUID. |
| bytes | Bytes. |
| str | Null-terminated ANSI character string with maximum length of 8000 characters. |
| wstr | Null-terminated Unicode character string with maximum length of 8000 characters. |
| numeric | Numeric. |
| dbDate | A structure that consists of year, month, day. |
| dateTime | Date time. |
| dbTime | A structure that consists of hour, minute, second. |
| dbTimeStamp | A structure that consists of year, month, day, hour, minute, seconds, and fractional parts. |
| image | Image. |
| text | Text. |
| nText | A Unicode character string with a maximum length of $2^{30} - 1$ characters. |
| dbTime2 | A **dbtime** type that also includes fractional seconds. |
| dbTimeStamp2 | A **dbTimeStamp** type that also includes fractional seconds. |
| dbTimeStampOffset | A **dbTimeStamp** type that has a time zone offset. |

### 2.7.1.1.1.1.6.7 PipelineComponentPropertyAttributeGroup

The **PipelineComponentPropertyAttributeGroup** attribute group specifies the attributes that are used for all input and output columns that are specified by components.

The following is the XSD for the **PipelineComponentPropertyAttributeGroup** attribute group.

```
    <xs:attributeGroup name="PipelineComponentPropertyAttributeGroup">
        <xs:attribute name="dataType"
                    type="DTS:PipelineComponentDataTypeEnum"
                    form="unqualified"/>
        <xs:attribute name="name"
                    type="xs:string"
                    form="unqualified"/>
        <xs:attribute name="state"
                    type="DTS:PipelineComponentStateEnum"
                    form="unqualified" use="optional" default="default"/>
        <xs:attribute name="isArray" type="xs:boolean"
                    use="optional" default="false" form="unqualified"/>
        <xs:attribute name="description" type="xs:string"
                    use="optional" default="" form="unqualified"/>
        <xs:attribute name="typeConverter" type="xs:string"
                    use="optional" default="" form="unqualified"/>
        <xs:attribute name="UITypeEditor" type="xs:string"
                    use="optional" default="" form="unqualified"/>
        <xs:attribute name="containsID" type="xs:boolean"
                    use="optional" default="false" form="unqualified"/>
        <xs:attribute name="expressionType"
                    type="DTS:PipelineComponentExpressionTypeEnum"
                    use="optional" default="None" form="unqualified"/>
    </xs:attributeGroup>
```

The following table specifies additional information for the attributes of the **PipelineComponentPropertyAttributeGroup** attribute group.

| Attribute | Description |
|---|---|
| name | A string value that specifies the name for this component **property** element. |
| dataType | An enumeration value that specifies the data type that is contained in the value for the **property** element. |
| state | An enumeration value that specifies the manner in which the **property** element value is serialized. |
| isArray | A Boolean value that specifies whether the value of the **property** element is an array.<br>TRUE specifies that the value of the **property** element is an array.<br>FALSE specifies that the value of the **property** element is not an array. |
| description | A string value that specifies the description for this component **property** element. |
| typeConverter | A string value that specifies the class name of a class that inherits from **System.ComponentModel.TypeConverter**. This class can be called to do type conversion from this component property type to another type. |
| UITypeEditor | A string value that specifies the class name of a class that derives from **System.Drawing.Design.UITypeEditor** and can be called by a user interface to edit a component property. |
| containsID | A Boolean value that specifies whether this property value is a reference to a column.<br>TRUE specifies that this property value is a reference to a column.<br>FALSE specifies that this property value is not a reference to a column. |

| Attribute | Description |
|---|---|
| expressionType | An enumeration value that specifies whether the value for the property can be specified through an expression. |

## 2.7.1.1.2 PipelinePathsType

The **PipelinePathsType** complex type specifies a collection of path objects for the Pipeline Task executable.

The following is the XSD for the **PipelinePathsType** complex type.

```
<xs:complexType name="PipelinePathsType">
  <xs:sequence>
    <xs:element name="path" type="DTS:PipelinePathType"
        form="unqualified" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

The following table specifies the elements of the **PipelinePathsType** complex type.

| Element | Type definition | Description |
|---|---|---|
| path | PipelinePathType | A complex type that specifies a single path within a pipeline. |

## 2.7.1.1.2.1 PipelinePathType

The **PipelinePathType** complex type specifies a single path between two components of a Pipeline Task executable.

The following is the XSD for the **PipelinePathType** complex type.

```
<xs:complexType name="PipelinePathType">
  <xs:sequence/>
  <xs:attribute name="refId" type="xs:string" use="required" form="unqualified"/>
  <xs:attribute name="id" type="xs:string" form="unqualified"/>
  <xs:attribute name="name" type="xs:string" form="unqualified"/>
  <xs:attribute name="description" type="xs:string" form="unqualified"/>
  <xs:attribute name="startId" type="xs:string" form="unqualified"/>
  <xs:attribute name="endId" type="xs:string" form="unqualified"/>
</xs:complexType>
```

The following table specifies the attributes of the **PipelinePathType** complex type.

| Attribute | Description |
|---|---|
| refId | A string value that specifies a unique **refId** for this object. Elsewhere within the containing executable or within the package, the value of **refId** can be used to refer to this object.<64> |
| id | A string value that specifies an identifier for this **path** element. Other elements can refer to this **path** element by using this **id** value. |
| name | A string value that specifies the name for this **path** element. |

| Attribute | Description |
|---|---|
| description | A string value that specifies the description for this **path** element. |
| startId | A string value that specifies the identifier for a component that is where the path that this **path** element describes starts. |
| endId | A string value that specifies the identifier for a component that is where the path that this **path** element describes ends. |

### 2.7.1.2 ActiveXScriptTaskData Element

*Applies to DTSX2 schema for version 2012/01 (DTSX2 2012/01)*

The **ActiveXScriptTaskData** element is used to specify parameters, properties, and settings that are specific to the ActiveX Script Task. The **ActiveXScriptTaskData** element resides in the ActiveXScriptTask namespace. The ExecutableObjectDataType type refers to the **ActiveXScriptTaskData** element within an XSD **choice** element by using the XSD **ref** attribute, as in the following XSD fragment.

```
<xs:element ref="ActiveXScriptTask:ActiveXScriptTaskData" />
```

### 2.7.1.2.1 ActiveXScriptTask Namespace

*Applies to DTSX2 schema for version 2012/01 (DTSX2 2012/01)*

The **ActiveXScriptTask** namespace specifies elements and types that are used by the ActiveX Script Task.

The following is the XSD declaration for the **ActiveXScriptTask** namespace.

```
xmlns:ActiveXScriptTask=
"www.microsoft.com/sqlserver/dts/tasks/activexscripttask"
```

### 2.7.1.2.1.1    ActiveXScriptTaskData Element

*Applies to DTSX2 schema for version 2012/01 (DTSX2 2012/01)*

The **ActiveXScriptTaskData** element is used to specify parameters, properties, and settings that are specific to the ActiveX Script Task.

The following is the XSD for element declaration for the **ActiveXScriptTaskData** element in the ActiveXScriptTask namespace.

```
<xs:element name="ActiveXScriptTaskData"
            type="ActiveXScriptTask:ActiveXScriptTaskDataType"/>
```

### 2.7.1.2.1.1.1 ActiveXScriptTaskDataType

*Applies to DTSX2 schema for version 2012/01 (DTSX2 2012/01)*

The **ActiveXScriptTaskDataType** complex type holds parameters, properties, and settings for an ActiveX Script Task.

The following is the XSD for the **ActiveXScriptTaskDataType** complex type.

```
<xs:complexType name="ActiveXScriptTaskDataType">
  <xs:sequence/>
  <xs:attribute name="ScriptingLanguage"
                type="ActiveXScriptTask:ScriptingLanguageEnum"/>
  <xs:attribute name="ScriptText" type="xs:string"/>
  <xs:attribute name="EntryMethod" type="xs:string"/>
</xs:complexType>
```

The **ActiveXScriptTaskDataType** complex type MUST NOT contain any elements or values.

The following table provides additional information for the attributes of the **ActiveXScriptTaskDataType** complex type.

| Attribute | Description |
|---|---|
| ScriptingLanguage | This enumeration value specifies the language the script is written in. |
| ScriptText | This string value contains the software code for the script in the language that is indicated by the **ScriptingLanguage** attribute. |
| EntryMethod | This string value specifies the name of the method where the script is entered. |

#### 2.7.1.2.1.1.1.1 ScriptingLanguageEnum

The ScriptingLanguageEnum type contains the enumeration values for the ScriptingLanguage attribute.

The following is the XSD for the ScriptingLanguageEnum type.

```
<xs:simpleType name="ScriptingLanguageEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="VBScript"/>
    <xs:enumeration value="JScript.Compact"/>
    <xs:enumeration value="JScript"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the ScriptingLanguageEnum type.

| Enumeration value | Description |
|---|---|
| VBScript | Visual Basic Script language [MSDN-VBSLR]. |
| JScript.Compact | Java Script Compact EMCA 327 [ECMA-327]. |
| Jscript | Java Script language [JavaSpec]. |

### 2.7.1.3 ASDdlDataObjectDataType

The **ASDdlDataObjectDataType** complex type is used to specify parameters, properties, and settings that are specific to the Analysis Services (AS) Execute Data Definition Language (DDL) Task.

The following is the XSD for the **ASDdlDataObjectDataType** complex type.

```
<xs:complexType name="ASDdlDataObjectDataType">
  <xs:sequence/>
  <xs:attribute name="ConnectionName" type="xs:string" form="unqualified"/>
  <xs:attribute name="SourceType" form="unqualified" type="DTS:SourceTypeEnum"/>
  <xs:attribute name="Source" type="xs:string" form="unqualified"/>
</xs:complexType>
```

The **ASDdlDataObjectDataType** complex type MUST NOT contain any elements or values.

The following table specifies the attributes of the **ASDdlDataObjectDataType** complex type.

| Attribute | Description |
|---|---|
| ConnectionName | The name of a connection manager that connects to an instance of Analysis Services or to an Analysis Services project. |
| SourceType | Specifies the source of the DDL statements. |
| Source | If the **SourceType** is "variable", this string contains the name of the variable that contains the DDL statements that the task executes.<br><br>If the **SourceType** is "DirectInput", this string contains the text of the DDL statement as encoded XML.<br><br>If the **SourceType** is "FileConnection", this string contains the value of the **ObjectName** property of a connection manager. |

## 2.7.1.4 ASProcessingDataObjectDataType

The **ASProcessingDataObjectDataType** complex type is used to specify parameters, properties, and settings that are specific to the AS Processing Task.

The following is the XSD for the **ASProcessingDataObjectDataType** complex type.

```
<xs:complexType name="ASProcessingDataObjectDataType">
  <xs:sequence/>
  <xs:attribute name="ConnectionName" type="xs:string" form="unqualified"/>
  <xs:attribute name="ProcessingCommands" type="xs:string" form="unqualified"/>
</xs:complexType>
```

The **ASProcessingDataObjectDataType** type MUST NOT contain any elements or values.

The following table provides additional information about the attributes of the **ASProcessingDataObjectDataType** type.

| Attribute | Description |
|---|---|
| ConnectionName | This string contains the name of a **ConnectionManager** that is available in this package. The **ConnectionName** attribute is matched to the **ObjectName** property of the **ConnectionManager** type. |
| ProcessingCommands | This string contains a stream of encoded XML that contains the actual processing commands that are to be executed. |

### 2.7.1.5 BulkInsertTaskData Element

The **BulkInsertTaskData** element is used to specify parameters, properties, and settings that are specific to the Bulk Insert Task. The **BulkInsertTaskData** element resides in the BulkInsertTask namespace. The **BulkInsertTaskData** type refers to the **BulkInsertTaskData** element within an XSD **choice** element, using the XSD **ref** attribute, as in the following XSD fragment.

The following is the XSD for the **BulkInsertTaskData** element.

```
<xs:element ref="BulkInsertTask:BulkInsertTaskData"/>
```

For more information about the **BulkInsertTaskData** element, see ExecutableObjectDataType.

### 2.7.1.5.1 BulkInsertTask Namespace

The **BulkInsertTask** namespace specifies the elements and types that are used by the Bulk Insert Task.

The following is the XSD declaration for the **BulkInsertTask** namespace.

```
xmlns:BulkInsertTask=
"www.microsoft.com/sqlserver/dts/tasks/bulkinserttask"
```

### 2.7.1.5.1.1 BulkInsertTaskData Element

The **BulkInsertTaskData** element is used to specify parameters, properties, and settings that are specific to the Bulk Insert Task.

The following is the XSD for the element declaration for the **BulkInsertTaskData** element in the BulkInsertTask namespace.

```
<xs:element name="BulkInsertTaskData"
            type="BulkInsertTask:BulkInsertTaskDataType"/>
```

### 2.7.1.5.1.2 BulkInsertTaskDataType

The **BulkInsertTaskDataType** complex type holds parameters, properties, and settings for a Bulk Insert Task.

The following is the XSD for the **BulkInsertTaskDataType** complex type.

```
<xs:complexType name="BulkInsertTaskDataType">
  <xs:sequence/>
  <xs:attribute name="BatchSize" type="xs:int" use="optional" default="0"/>
  <xs:attribute name="CheckConstraints" use="optional" default="True">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="True"/>
        <xs:enumeration value="False"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="CodePage" type="BulkInsertTask:CodePageType"
                use="optional" default="RAW"/>
  <xs:attribute name="SourceConnectionName" type="xs:string"
                use="optional" default=""/>
  <xs:attribute name="DestinationConnectionName" type="xs:string"
                use="optional" default=""/>
```

```
        <xs:attribute name="DataFileType"
                      type="BulkInsertTask:DataFileTypeEnum"
                      use="optional" default="DTSBulkInsert_DataFileType_Char"/>
        <xs:attribute name="DestinationTableName" type="xs:string"
                      use="optional" default=""/>
        <xs:attribute name="FirstRow" type="xs:int" use="optional" default="1"/>
        <xs:attribute name="LastRow" type="xs:int" use="optional" default="0"/>
        <xs:attribute name="UseFormatFile" use="optional" default="False">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="True"/>
            <xs:enumeration value="False"/>
          </xs:restriction>
        </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="FormatFile" type="xs:string" use="optional" default=""/>
        <xs:attribute name="SortedData" type="xs:string" use="optional" default=""/>
        <xs:attribute name="TableLock" use="optional" default="False">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="True"/>
              <xs:enumeration value="False"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="KeepIdentity" use="optional" default="False">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="True"/>
              <xs:enumeration value="False"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="KeepNulls" use="optional" default="False">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="True"/>
              <xs:enumeration value="False"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="FieldTerminator"
  type="BulkInsertTask:BulkInsertTaskTerminatorEnum" use="optional" default="Tab"/>
        <xs:attribute name="RowTerminator" type="BulkInsertTask:BulkInsertTaskTerminatorEnum"
  use="optional" default="{CR}{LF}"/>
        <xs:attribute name="FireTriggers" use="optional" default="False">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="True"/>
              <xs:enumeration value="False"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="MaximumErrors" type="xs:int" use="optional" default="0"/>
    </xs:complexType>
```

The **BulkInsertTaskDataType** complex type MUST NOT contain elements or values.

The following table specifies additional information for the attributes of the **BulkInsertTaskDataType** complex type.

| Attribute | Description |
|---|---|
| BatchSize | This integer specifies the number of rows in a batch. Each batch is processed as a single transaction.<65> |
| CheckConstraints | This enumeration value specifies whether constraints on the table are checked |

| Attribute | Description |
|---|---|
| | during the bulk insert operation. |
| | "True" specifies that constraints are checked. |
| | "False" specifies that constraints are not checked. |
| CodePage | This value specifies the code page of the data in the data file. |
| SourceConnectionName | This string specifies the name of the source connection file. |
| DestinationConnectionName | This string specifies the name of the connection to the destination database. |
| DataFileType | This enumeration value specifies the type of text that is in the file. |
| DestinationTableName | This string specifies the name of the table in the destination database. |
| FirstRow | The row in the data file at which insertion begins. |
| LastRow | The row in the data file at which insertion ends.<66> |
| UseFormatFile | This enumeration value specifies whether a format file is used for the bulk insert operation.<br>"True" specifies that a format file is used for the bulk insert operation.<br>"False" specifies that a format file is not used for the bulk insert operation. |
| FormatFile | The string specifies the name of the format file. This string is used only when the **UseFormatFile** attribute is set to "True". |
| SortedData | This string specifies the columns on which to sort the data. |
| TableLock | This enumeration value specifies whether a table level lock is acquired for the bulk insert operation.<br>"True" specifies that a table level lock is acquired for the bulk insert operation.<br>"False" specifies that a table level lock is not acquired for the bulk insert operation. |
| KeepIdentity | This enumeration value specifies whether to keep data values from the data file for values for identity columns.<br>"True" specifies that values present in the data file are kept for identity columns.<br>"False" specifies that values present in the data file are not kept for identity columns. |
| KeepNulls | This enumeration value specifies whether null values in the input are retained.<br>"True" specifies that null values in the input are retained.<br>"False" specifies that a default value is used if a null value is encountered in the input. |
| FieldTerminator | This enumeration value specifies the character that terminates a field. |
| RowTerminator | This enumeration value specifies the character that terminates a row. |
| FireTriggers | This enumeration value specifies whether insert triggers that are specified for the table are fired during the bulk insert operation.<br>"True" specifies that triggers are fired during the bulk insert operation.<br>"False" specifies that triggers are not fired during the bulk insert operation. |
| MaximumErrors | This enumeration value specifies the maximum number of errors allowed. |

### 2.7.1.5.1.2.1 DataFileTypeEnum

The DataFileTypeEnum type contains the enumeration values for the DataFileType attribute.

The following is the XSD for the DataFileTypeEnum type.

```
<xs:simpleType name="DataFileTypeEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="DTSBulkInsert_DataFileType_Char"/>
    <xs:enumeration value="DTSBulkInsert_DataFileType_Native"/>
    <xs:enumeration value="DTSBulkInsert_DataFileType_WideChar"/>
    <xs:enumeration value="DTSBulkInsert_DataFileType_WideNative"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the DataFileTypeEnum type.

| Enumeration value | Description |
|---|---|
| DTSBulkInsert_DataFileType_Char | Specifies that the data file contains character data. |
| DTSBulkInsert_DataFileType_Native | Performs the bulk copy by using character types that are native to the database.<67> |
| DTSBulkInsert_DataFileType_WideChar | Specifies that the data file contains Unicode [UNICODE] characters. |
| DTSBulkInsert_DataFileType_WideNative | Functions the same as DTSBulkInsert_DataFileType_Native, except that columns of type char, varchar, and text are stored as Unicode.<68> |

### 2.7.1.5.1.2.2 CodePageType

The **CodePageType** type is used to specify the value of the code page of the data in the data file.

The following is the XSD for the **CodePageType** type.

```
<xs:simpleType name="CodePageType">
  <xs:union memberTypes="xs:int">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="ACP"/>
        <xs:enumeration value="OEM"/>
        <xs:enumeration value="RAW"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:union>
</xs:simpleType>
```

The **CodePageType** type is a union that allows an integer value or a value from a string enumeration. If the value is an integer, it represents the ANSI code page. If the value is a string enumeration value, it is one of the values in the following table.

| Enumeration value | Description |
|---|---|
| ACP | Columns of **char**, **varchar**, or **text** data type SHOULD be converted from the ANSI code page (ISO 1252 [ISO/IEC8859-1]) to the server code page.<69> |
| OEM | Columns of **char**, **varchar**, or **text** data type SHOULD be converted from the system OEM |

| Enumeration value | Description |
|---|---|
| | code page to the server code page.<70> |
| RAW | No conversion from one code page to another occurs. |

### 2.7.1.5.1.2.3 BulkInsertTaskTerminatorEnum

The **BulkInsertTaskTerminatorEnum** type contains the enumeration values that can be specified for the demarcation of fields or records in the input data file.

The following is the XSD for the **BulkInsertTaskTerminatorEnum** type.

```
<xs:simpleType name="BulkInsertTaskTerminatorEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="{CR}{LF}"/>
    <xs:enumeration value="{CR}"/>
    <xs:enumeration value="{LF}"/>
    <xs:enumeration value="Semicolon {;}"/>
    <xs:enumeration value="Comma {,}"/>
    <xs:enumeration value="Tab"/>
    <xs:enumeration value="Vertical Bar {|}"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **BulkInsertTaskTerminatorEnum** type.

| Enumeration value | Description |
|---|---|
| {CR}{LF} | Specifies that the terminator is a carriage return character followed by a line feed character. |
| {CR} | Specifies that the terminator is a carriage return character. |
| {LF} | Specifies that the terminator is a line feed character. |
| Semicolon {;} | Specifies that the terminator is a semicolon character. |
| Comma {,} | Specifies that the terminator is a comma character. |
| Tab | Specifies that the terminator is a tab character. |
| Vertical Bar {|} | Specifies that the terminator is a vertical bar character. |

### 2.7.1.6 DMQueryTaskDataObjectDataType

The **DMQueryTaskDataObjectDataType** complex type is used to specify parameters, properties, and settings that are specific to the Data Mining Query Task.

The following is the XSD for the **DMQueryTaskDataObjectDataType** complex type.

```
<xs:complexType name="DMQueryTaskDataObjectDataType">
  <xs:sequence>
    <xs:element name="QueryString" form="unqualified"
```

```
            type="DTS:DMQueryTaskQueryStringContainerType"/>
    <xs:element name="QueryBuilderQueryString" form="unqualified"
            type="DTS:DMQueryTaskQueryStringContainerType"/>
    <xs:element name="QueryBuilderSpecification" form="unqualified"
            type="DTS:DMQueryBuilderSpecificationType"/>
  </xs:sequence>
  <xs:attribute name="InputConnection" type="xs:string"
            form="unqualified"/>
  <xs:attribute name="OutputConnection" type="xs:string"
            form="unqualified"/>
  <xs:attribute name="ModelStructureName" type="xs:string"
            form="unqualified"/>
  <xs:attribute name="ModelName" type="xs:string" form="unqualified"/>
  <xs:attribute name="OverwriteOutputTable" form="unqualified">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="True"/>
        <xs:enumeration value="False"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="OutputTableName" type="xs:string" form="unqualified"/>
</xs:complexType>
```

The following table specifies the elements of the **DMQueryTaskDataObjectDataType** complex type.

| Element | Type definition | Description |
|---------|-----------------|-------------|
| QueryString | DMQueryTaskQueryStringContainerType | A string that specifies a data mining query.<71> |
| QueryBuilderQueryString | DMQueryTaskQueryStringContainerType | A string that specifies a data mining query that appears in the UI for data mining as the initial query string, which the user can then modify. |
| QueryBuilderSpecification | DMQueryBuilderSpecificationType | The complex content contained in the **QueryBuilderSpecification** element specifies information about the state of the Integration Services UI between sessions so that editing sessions can start from where a previous session finished. Further description of the contents of this element is not within the scope of this document. |

The following table specifies the attributes of the **DMQueryTaskDataObjectDataType** complex type.

| Attribute | Description |
|-----------|-------------|
| InputConnection | This string value is the connection string to the data source for the data mining operation. |
| OutputConnection | This string value is the connection string to the database where the results of the data mining operation are written. |
| ModelStructureName | This string value contains the name of the data mining structure [MSDN-MSASDM] that is used for the data mining operation. |
| ModelName | This string value contains the name of the data mining model [MSDN-MMASDM] that is used for the data mining operation. |
| OverwriteOutputTable | This enumeration value specifies whether the table that is specified in the **OutputTableName** attribute can be overwritten. |

| Attribute | Description |
|---|---|
| | "True" specifies that data in the output table can be overwritten.<br>"False" specifies that data in the output table cannot be overwritten. |
| OutputTableName | The name of the table in the output database where the results of the data mining operation are written. |

### 2.7.1.6.1 DMQueryTaskQueryStringContainerType

The **DMQueryTaskQueryStringContainerType** complex type contains a string value that is the value of the specific element that is of the **DMQueryTaskQueryStringContainerType** complex type.

The following is the XSD for the **DMQueryTaskQueryStringContainerType** complex type.

```
<xs:complexType name="DMQueryTaskQueryStringContainerType">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="Sensitive" type="xs:boolean" form="unqualified"/>
      <xs:attribute name="Encrypted" type="xs:boolean" form="unqualified"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

The **DMQueryTaskQueryStringContainerType** complex type MUST NOT contain any elements. The value of the **DMQueryTaskQueryStringContainerType** complex type is a string value because the type is an extension of the string type. Attributes are added to the string type, as specified in this section.

The following table specifies additional information for the attributes of the **DMQueryTaskQueryStringContainerType** complex type.

| Attribute | Description |
|---|---|
| Sensitive | The attribute value contains an indicator for whether the string that is the value of the element is treated as sensitive.<br>FALSE specifies that the element string value is not considered to be sensitive.<br>TRUE specifies that the element string value is considered to be sensitive. |
| Encrypted | The attribute value contains an indicator for whether the element value is encrypted.<br>FALSE specifies that the element value is not encrypted.<br>TRUE specifies that the element value is encrypted.<br>For more details about the encryption algorithm that is used if the value of the **Encrypted** attribute is TRUE, see Security. |

### 2.7.1.6.2 DMQueryBuilderSpecificationType

The **DMQueryBuilderSpecificationType** complex type contains unspecified XML content that is used for the Integration Services UI.

The following is the XSD for the **DMQueryBuilderSpecificationType** complex type.

```
<xs:complexType name="DMQueryBuilderSpecificationType">
  <xs:sequence>
```

```
          <xs:any minOccurs="0" maxOccurs="unbounded"
                 namespace="##any" processContents="lax"/>
      </xs:sequence>
      <xs:attribute name="Sensitive" type="xs:boolean" form="unqualified"/>
      <xs:attribute name="Encrypted" type="xs:boolean" form="unqualified"/>
  </xs:complexType>
```

The following table specifies additional information for the attributes of the
**DMQueryBuilderSpecificationType** complex type.

| Attribute | Description |
|-----------|-------------|
| Sensitive | The attribute value contains an indicator for whether the element content is treated as sensitive. FALSE specifies that the element content is not considered to be sensitive. TRUE specifies that the element content is considered to be sensitive. |
| Encrypted | The attribute value contains an indicator for whether the element content is encrypted. FALSE specifies that the element content is not encrypted. TRUE specifies that the element content is encrypted. |

### 2.7.1.7  DataProfilingTaskDataObjectDataType

The **DataProfilingTaskDataObjectDataType** complex type is used to specify parameters,
properties, and settings that are specific to the Data Profiling Task.

The following is the XSD for the **DataProfilingTaskDataObjectDataType** complex type.

```
    <xs:complexType name="DataProfilingTaskDataObjectDataType">
      <xs:sequence>
        <xs:element name="ProfileInput" type="xs:string" form="unqualified"/>
      </xs:sequence>
      <xs:attribute name="DestinationType" form="unqualified"
                    type="DTS:DestinationTypeEnum"
                    use="optional" default="FileConnection"/>
      <xs:attribute name="Destination" type="xs:string"
                    form="unqualified" use="optional" default=""/>
      <xs:attribute name="OverwriteDestination" form="unqualified"
                    use="optional" default="False">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="True"/>
            <xs:enumeration value="False"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:complexType>
```

The following table specifies the elements of the **DataProfilingTaskDataObjectDataType** complex
type.

| Element | Type definition | Description |
|---------|-----------------|-------------|
| ProfileInput | String. Content of the string is further defined in Data Profiling ProfileInput String Format Definition. | A string value that contains encoded XML that specifies parameters, properties, and settings for the Data Profiling Task. |

The following table specifies the attributes of the **DataProfilingTaskDataObjectDataType** complex type.

| Attribute | Description |
|-----------|-------------|
| DestinationType | This enumeration value specifies whether to save the output to a file connection or to a variable. |
| Destination | This string specifies the name of the file connection or the name of the variable in which to save the Data Profiling Task output. |
| OverwriteDestination | This enumeration value specifies whether the output from this task can overwrite any previously existing content at the output destination.<br>"True" specifies that previously existing content at the destination for output can be overwritten.<br>"False" specifies that previously existing content at the destination for output cannot be overwritten. |

### 2.7.1.7.1 Data Profiling ProfileInput String Format Definition

The **ProfileInput** element appears within the Integration Services package stream as a string value. This string value is shown in the XSD in DataProfilingTaskDataObjectDataType where the **ProfileInput** element is defined as type **xs:string**.

However, the content of that string value is encoded XML. This section specifies the XML in the string value by using XSD. Despite the use of XSD in these sections, this value is a string at the Integration Services package level.

#### 2.7.1.7.1.1 DataProfile Root Element of the ProfileInput String

The root element of the XML that is encoded as a string that is contained in the **ProfileInput** element is the **DataProfile** element.

The following is the XSD of the **DataProfile** element declaration.

```
<xs:element name="DataProfile">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="DataSources" type="DataSourcesType" />
      <!-- Input -->
      <xs:element name="DataProfileInput" type="DataProfileInputType"
                  minOccurs="0" maxOccurs="1" />
      <!-- Output -->
      <xs:element name="DataProfileOutput" type="DataProfileOutputType"
                  minOccurs="0" maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

The following table specifies additional information about the elements and types for the **DataProfile** root element.

| Element | Type definition | Description |
|---------|-----------------|-------------|
| DataSources | DataSourcesType | This complex type specifies the data sources that are referenced by the Data Profiling Task. |
| DataProfileInput | DataProfileInputType | This complex type specifies what to profile and how to profile it. |
| DataProfileOutput | DataProfileOutputType | This complex type specifies the profile output. |

### 2.7.1.7.1.1.1 DataSourcesType

The **DataSourcesType** complex type MAY contain references to data sources.<72>

The following is the XSD of the **DataSourcesType** complex type.

```
<xs:complexType name="DataSourcesType">
<xs:sequence>
</xs:sequence>
</xs:complexType>
```

### 2.7.1.7.1.1.2 DataProfileInputType

The **DataProfileInputType** complex type is used to specify parameters, properties, and settings that are specific to the Data Profiling Task.

The following is the XSD of the **DataProfileInputType** complex type.

```
<xs:complexType name="DataProfileInputType">
  <xs:sequence>
    <xs:element name="ProfileMode" type="ProfileModeType"
                minOccurs="1" maxOccurs="1" />
    <xs:element name ="Timeout" type="xs:int" minOccurs="0"
                maxOccurs ="1" default ="0" />
    <xs:element name="Requests" minOccurs="1" maxOccurs="1">
      <xs:complexType>
        <xs:sequence>
          <xs:element ref="ProfileRequest" minOccurs="0" maxOccurs="unbounded" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

The following table specifies additional information about the elements and types for the **DataProfileInputType** complex type.

| Element | Type definition | Description |
|---------|-----------------|-------------|
| ProfileMode | ProfileModeType | An enumeration value that specifies in which mode the profile is run.<73> |
| Timeout | xs:int | An integer value that specifies the timeout interval for the Data Profiling Task in seconds. |
| Requests | ProfileRequest | A complex type that contains all of the profile requests to be run by the Data Profiling Task. |

### 2.7.1.7.1.1.2.1    ProfileModeType

The **ProfileModeType** type specifies the enumeration values for the **ProfileMode** element.

The following is the XSD of the **ProfileModeType** type.

```
<xs:simpleType name="ProfileModeType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Fast" />
    <xs:enumeration value="Exact" />
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **ProfileModeType** type.

| Enumeration value | Description |
|---|---|
| Fast | A profile that is computed in fast mode and does not always return an exact profile. |
| Exact | A profile that is computed and always returns an exact profile. |

### 2.7.1.7.1.1.2.2    ProfileRequest Element

The **ProfileRequest** element is an abstract element that is a placeholder for elements that contain a specific data profile request. Elements that can be substituted for this placeholder have the **substitutionGroup** attribute of the element set to the value "ProfileRequest".

The following is the XSD of the **ProfileRequest** element.

```
<xs:element name="ProfileRequest" type="ProfileRequestType" abstract="true" />
```

### 2.7.1.7.1.1.2.2.1    ProfileRequestType

The **ProfileRequestType** abstract type is used to hold a profile request. The **ProfileRequestType** abstract type specifies an attribute that is common to many requests. The content of elements of type **ProfileRequestType** is specified for each specific element that can be substituted for the element of this type that appears in an XML instance document.

The following is the XSD of the **ProfileRequestType** abstract type.

```
<xs:complexType name="ProfileRequestType" abstract="true">
  <xs:attribute name="ID" type="xs:string" use="required" />
</xs:complexType>
```

The **ProfileRequestType** abstract type can contain elements, depending on the type of the element that is substituted.

The following table specifies additional information for the attributes of the **ProfileRequestType** abstract type.

| Attribute | Description |
|-----------|-------------|
| ID | A string value that specifies an identification string that is used to refer to the data profile request elsewhere in the package. |

#### 2.7.1.7.1.1.2.2.2 Elements That Can Be Substituted for the ProfileRequest Element

The following elements have the XSD **substitutionGroup** attribute set to "ProfileRequest" in their XSD definition. Thus, they can be substituted for the ProfileRequest element. More details about the following elements and their associated complex types are provided later in this section.

- ColumnStatisticsProfileRequest

- ColumnLengthDistributionProfileRequest

- ColumnValueDistributionProfileRequest

- ColumnNullRatioProfileRequest

- ColumnPatternProfileRequest

- CandidateKeyProfileRequest

- FunctionalDependencyProfileRequest

- InclusionProfileRequest

##### 2.7.1.7.1.1.2.2.2.1 ColumnStatisticsProfileRequest Element

The **ColumnStatisticsProfileRequest** element is used to specify parameters, properties, and settings that are used to request a column statistics profile. A column statistics profile reports statistics about numeric or date/time columns of a database.

The **ColumnStatisticsProfileRequest** element has the XSD **substitutionGroup** attribute set to the value "ProfileRequest"; therefore, it is one of the elements that can be substituted for the **ProfileRequest** element.

The following is the XSD of the **ColumnStatisticsProfileRequest** element declaration.

```
<xs:element name="ColumnStatisticsProfileRequest"
            type="ColumnStatisticsProfileRequestType"
            substitutionGroup="ProfileRequest" />
```

###### 2.7.1.7.1.1.2.2.2.1.1 ColumnStatisticsProfileRequestType

The **ColumnStatisticsProfileRequestType** complex type is the type of the **ColumnStatisticsProfileRequest** element.

The following is the XSD of the **ColumnStatisticsProfileRequestType** complex type.

```
<xs:complexType name="ColumnStatisticsProfileRequestType"
                final="#all">
  <xs:complexContent>
    <xs:extension base="ColumnProfileRequestType">
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

All of the elements and attributes of the **ColumnStatisticsProfileRequestType** type are contained in the definition of its base type, the **ColumnProfileRequestType** complex type. The **ColumnStatisticsProfileRequestType** type MUST NOT contain additions or extensions that are not specified in the **ColumnProfileRequestType** complex type.

#### 2.7.1.7.1.1.2.2.2.2 ColumnLengthDistributionProfileRequest Element

The **ColumnLengthDistributionProfileRequest** element is used to specify parameters, properties, and settings that are used to request column length distribution profiles. A column length distribution profile, an option of the Data Profiling Task, reports all the distinct lengths of string values in the selected column and the percentage of rows in the table that each length represents.

The **ColumnLengthDistributionProfileRequest** element has the XSD **substitutionGroup** attribute set to the value "ProfileRequest"; therefore, it is one of the elements that can be substituted for the **ProfileRequest** element.

The following is the XSD of the **ColumnLengthDistributionProfileRequest** element declaration.

```
<xs:element name="ColumnLengthDistributionProfileRequest"
            type="ColumnLengthDistributionProfileRequestType"
            substitutionGroup="ProfileRequest" />
```

#### 2.7.1.7.1.1.2.2.2.2.1 ColumnLengthDistributionProfileRequestType

The **ColumnLengthDistributionProfileRequestType** complex type is the type of the **ColumnLengthDistributionProfileRequest** element.

The following is the XSD of the **ColumnLengthDistributionProfileRequestType** complex type.

```
<xs:complexType name="ColumnLengthDistributionProfileRequestType" final="#all">
  <xs:complexContent>
    <xs:extension base="ColumnProfileRequestType">
      <xs:sequence>
        <xs:element name="IgnoreLeadingSpace" type="xs:boolean"
                    minOccurs="0" maxOccurs="1" default="false" />
        <xs:element name="IgnoreTrailingSpace" type="xs:boolean"
                    minOccurs="0" maxOccurs="1" default ="true" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The **ColumnLengthDistributionProfileRequestType** complex type is an extension of the ColumnProfileRequestType complex type. The following table specifies additional information about the elements in the **ColumnLengthDistributionProfileRequestType** complex type.

| Element | Type definition | Description |
|---------|-----------------|-------------|
| IgnoreLeadingSpace | xs:boolean | A Boolean value that specifies whether to ignore leading spaces when comparing string values. |
| | | TRUE indicates that leading spaces MUST be ignored when comparing string values. |
| | | FALSE indicates that leading spaces MUST NOT be ignored when comparing string values. |
| IgnoreTrailingSpace | xs:boolean | A Boolean value that specifies whether to ignore trailing spaces when comparing string values. |
| | | TRUE indicates that trailing spaces MUST be ignored when comparing |

| Element | Type definition | Description |
|---|---|---|
| | | string values. |
| | | FALSE indicates that trailing spaces MUST NOT be ignored when comparing string values. |

#### 2.7.1.7.1.1.2.2.2.3   ColumnValueDistributionProfileRequest Element

The **ColumnValueDistributionProfileRequest** element is used to specify parameters, properties, and settings that are used to request column value distribution profiles. A column value distribution profile, an option of the Data Profiling Task, reports all the distinct values in the selected column and the percentage of rows in the table that each value represents.

The **ColumnValueDistributionProfileRequest** element has the XSD **substitutionGroup** attribute set to the value "ProfileRequest"; therefore, it is one of the elements that can be substituted for the **ProfileRequest** element.

The following is the XSD of the **ColumnValueDistributionProfileRequest** element declaration.

```
<xs:element name="ColumnValueDistributionProfileRequest"
            type="ColumnValueDistributionProfileRequestType"
            substitutionGroup="ProfileRequest" />
```

#### 2.7.1.7.1.1.2.2.2.3.1   ColumnValueDistributionProfileRequestType

The **ColumnValueDistributionProfileRequestType** complex type is the type of the **ColumnValueDistributionProfileRequest** element.

The following is the XSD of the **ColumnValueDistributionProfileRequestType** complex type.

```
<xs:complexType name="ColumnValueDistributionProfileRequestType" final="#all">
  <xs:complexContent>
    <xs:extension base="ColumnProfileRequestType">
      <xs:sequence>
        <xs:element name="Option" type="ValueDistributionOptions" minOccurs = "1" />
        <xs:element name="FrequentValueThreshold" type="xs:double"
                    minOccurs="0" maxOccurs="1" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The **ColumnValueDistributionProfileRequestType** complex type is an extension of the ColumnProfileRequestType complex type. The following table specifies additional information about the elements in the **ColumnValueDistributionProfileRequestType** complex type.

| Element | Type definition | Description |
|---|---|---|
| Option | ValueDistributionOptions | An enumeration value that specifies whether to include all values in the distribution computation. |
| FrequentValueThreshold | xs:double | A double value between 0 and 1, inclusive, that specifies the threshold at which to include values in the distribution computation. This value is ignored if the value of the **Option** attribute is not equal to "FrequentValueThreshold".[74] |

#### 2.7.1.7.1.1.2.2.2.3.1.1   ValueDistributionOptions

The **ValueDistributionOptions** type specifies the enumeration values for the **Options** element.

The following is the XSD of the **ValueDistributionOptions** type.

```
<xs:simpleType name="ValueDistributionOptions">
  <xs:restriction base="xs:string">
    <xs:enumeration value="AllValues" />
    <xs:enumeration value="FrequentValues" />
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **ValueDistributionOptions** type.

| Enumeration value | Description |
|---|---|
| AllValues | The distribution is computed for all values. |
| FrequentValues | The distribution is computed only for values whose frequency exceeds the minimum value that is specified in the **FrequentValueThreshold** element. |

#### 2.7.1.7.1.1.2.2.2.4 ColumnNullRatioProfileRequest Element

The **ColumnNullRatioProfileRequest** element is used to specify parameters, properties, and settings that are used to request column null ratio distribution profiles. A column null ratio distribution profile, an option of the Data Profiling Task, reports the percentage of null values in the selected column.

The **ColumnNullRatioProfileRequest** element has the XSD **substitutionGroup** attribute set to the value "ProfileRequest"; therefore, it is one of the elements that can be substituted for the **ProfileRequest** element.

The following is the XSD of the **ColumnNullRatioProfileRequest** element declaration.

```
<xs:element name="ColumnNullRatioProfileRequest"
            type="ColumnNullRatioProfileRequestType"
            substitutionGroup="ProfileRequest" />
```

#### 2.7.1.7.1.1.2.2.2.4.1 ColumnNullRatioProfileRequestType

The **ColumnNullRatioProfileRequestType** complex type is the type of the **ColumnNullRatioProfileRequest** element.

The following is the XSD of the **ColumnNullRatioProfileRequestType** complex type.

```
<xs:complexType name="ColumnNullRatioProfileRequestType" final="#all">
  <xs:complexContent>
    <xs:extension base="ColumnProfileRequestType">
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

All of the elements and attributes of the **ColumnNullRatioProfileRequestType** complex type are contained in the definition of its base type, the ColumnProfileRequestType complex type. The **ColumnNullRatioProfileRequestType** type MUST NOT contain additions or extensions that are not specified in the **ColumnProfileRequestType** complex type.

#### 2.7.1.7.1.1.2.2.2.5 ColumnPatternProfileRequest Element

The **ColumnPatternProfileRequest** element is used to specify parameters, properties, and settings that are used to request column pattern profiles. A column pattern profile, an option of the Data Profiling Task, reports a set of regular expressions that cover the specified percentage of values in a string column.

The **ColumnPatternProfileRequest** element has the XSD **substitutionGroup** attribute set to the value "ProfileRequest"; therefore, it is one of the elements that can be substituted for the **ProfileRequest** element.

The following is the XSD of the **InclusionProfileRequest** element declaration.

```
<xs:element name="ColumnPatternProfileRequest" type="ColumnPatternProfileRequestType"
            substitutionGroup="ProfileRequest" />
```

#### 2.7.1.7.1.1.2.2.2.5.1   ColumnPatternProfileRequestType

The **ColumnPatternProfileRequestType** complex type is the type of the **ColumnPatternProfileRequest** element.

The following is the XSD of the **ColumnPatternProfileRequestType** complex type.

```
<xs:complexType name=" ColumnPatternProfileRequestType " final="#all">
  <xs:complexContent>
    <xs:extension base="ColumnProfileRequestType">
      <xs:sequence>
        <xs:element name="MaxNumberOfPatterns" type="xs:int" minOccurs="0"/>
        <xs:element name="PercentageDataCoverageDesired"
                    type="xs:int" minOccurs="0"/>
        <xs:element name="CaseSensitive" type="xs:boolean" minOccurs="0"/>
        <xs:element name="Delimiters" type="xs:string" minOccurs="0"/>
        <xs:element name="Symbols" type="xs:string" minOccurs="0"/>
        <xs:element name="TagDataSourceId" type="xs:string" minOccurs="0"/>
        <xs:element name="TagTableName" type="TableQNameType" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The **ColumnPatternProfileRequestType** complex type is an extension of the base type, the ColumnProfileRequestType complex type. The following table specifies additional information about the elements in the **ColumnPatternProfileRequestType** complex type.

| Element | Type definition | Description |
|---|---|---|
| MaxNumberOfPatterns | xs:int | An integer value that specifies the maximum number of patterns that are returned by the column pattern profile task.<75> |
| PercentageDataCoverageDesired | xs:int | An integer value that specifies the percentage of the data that is described by the patterns that are returned by the column pattern profile task.<76> |
| CaseSensitive | xs:boolean | A Boolean value that specifies whether to consider strings as case sensitive.<br>TRUE specifies that case MUST be considered in determining the pattern.<br>FALSE specifies that case MUST be ignored in determining the pattern.<77> |
| Delimiters | xs:string | A string value that specifies the characters that are |

| Element | Type definition | Description |
|---|---|---|
| | | treated as the equivalent of spaces between words when tokenizing text.<78> |
| Symbols | xs:string | A string value that specifies the symbols that are retained as part of patterns.<79> |
| TagDataSourceId | xs:string | A string value that contains an identifier for the tag table. |
| TagTableName | TableQNameType | A complex type that specifies information to reference the table that is used as the tag table. |

### 2.7.1.7.1.1.2.2.2.6  CandidateKeyProfileRequest Element

The **CandidateKeyProfileRequest** element is used to specify parameters, properties, and settings that are used to request candidate key profiles. A candidate key profile, an option of the Data Profiling Task, reports whether a column or set of columns is a key or an approximate key for a specified table.

The **CandidateKeyProfileRequest** element has the XSD **substitutionGroup** attribute set to the value "ProfileRequest"; therefore, the **CandidateKeyProfileRequest** element is one of the elements that can be substituted for the **ProfileRequest** element.

The following is the XSD of the **CandidateKeyProfileRequest** element declaration.

```
<xs:element name="CandidateKeyProfileRequest"
            type="CandidateKeyProfileRequestType"
            substitutionGroup="ProfileRequest" />
```

### 2.7.1.7.1.1.2.2.2.6.1  CandidateKeyProfileRequestType

The **CandidateKeyProfileRequestType** complex type is the type of the **CandidateKeyProfileRequest** element.

The following is the XSD of the **CandidateKeyProfileRequestType** complex type.

```
<xs:complexType name="CandidateKeyProfileRequestType" final="#all">
  <xs:complexContent>
    <xs:extension base="TableProfileRequestType">
      <xs:sequence>
        <xs:element name="KeyColumns"
                    type="ColumnParameterCollectionType"
                    minOccurs="1" maxOccurs="1" />
        <xs:element name="ThresholdSetting"
                    type="StrengthThresholdSettingType"
                    minOccurs="1" maxOccurs="1" />
        <xs:element name="KeyStrengthThreshold" type="xs:double"
                    minOccurs="0" maxOccurs="1" />
        <xs:element name="VerifyOutputInFastMode" type="xs:boolean"
                    default ="true"  minOccurs="0" maxOccurs="1" />
        <xs:element name ="MaxNumberOfViolations" type ="xs:int"
                    minOccurs ="0" maxOccurs ="1" default ="-1" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The **CandidateKeyProfileRequestType** complex type is an extension of the **TableProfileRequestType** complex type. The following table specifies additional information about the elements in the **CandidateKeyProfileRequestType** complex type.

| Element | Type definition | Description |
|---------|----------------|-------------|
| KeyColumns | ColumnParameterCollectionType | A complex type that specifies the column or columns to be profiled as key candidates. |
| ThresholdSetting | StrengthThresholdSettingType | An enumeration value that specifies what type of threshold to use in identifying a key candidate. |
| KeyStrengthThreshold | xs:double | A double value that specifies the minimum strength that is required to report a candidate key. This value is used only if the **ThresholdSetting** element is set to the value "Specified".[80] |
| VerifyOutputInFastMode | xs:boolean | A Boolean value that specifies whether the output verification step is performed in fast mode. TRUE specifies that output verification is performed in fast mode. FALSE specifies that output verification is not performed in fast mode.[81] |
| MaxNumberOfViolations | xs:int | An integer value that specifies the maximum number of candidate key violations to report in the output. This option is disabled when the **KeyStrengthThreshold** element is set to the value "Exact".[82] |

#### 2.7.1.7.1.1.2.2.2.2.7  FunctionalDependencyProfileRequest Element

The **FunctionalDependencyProfileRequest** element is used to specify parameters, properties, and settings that are used to request functional dependency profiles. A functional dependency profile, an option of the Data Profiling Task, reports the extent to which the values in one column (the dependent column) depend on the values in another column or set of columns (the determinant column).

The **FunctionalDependencyProfileRequest** element has the XSD **substitutionGroup** attribute set to the value "ProfileRequest"; therefore, it is one of the elements that can be substituted for the **ProfileRequest** element.

The following is the XSD of the **FunctionalDependencyProfileRequest** element declaration.

```
<xs:element name="FunctionalDependencyProfileRequest"
            type="FunctionalDependencyProfileRequestType"
            substitutionGroup="ProfileRequest" />
```

#### 2.7.1.7.1.1.2.2.2.2.7.1  FunctionalDependencyProfileRequestType

The **FunctionalDependencyProfileRequestType** complex type is the type of the **FunctionalDependencyProfileRequest** element.

The following is the XSD of the **FunctionalDependencyProfileRequestType** complex type.

```
<xs:complexType name="FunctionalDependencyProfileRequestType" final="#all">
  <xs:complexContent>
    <xs:extension base="TableProfileRequestType">
      <xs:sequence>
        <xs:element name="DeterminantColumns"
                    type="ColumnParameterCollectionType"
                    minOccurs="1" maxOccurs="1" />
        <xs:element name="DependentColumn" type="ColumnParameterType"
                    minOccurs="1" maxOccurs="1" />
        <xs:element name="ThresholdSetting"
                    type="StrengthThresholdSettingType"
```

```
                    minOccurs="1" maxOccurs="1" />
        <xs:element name="FDStrengthThreshold" type="xs:double"
                    minOccurs="0" maxOccurs="1" />
        <xs:element name="VerifyOutputInFastMode" type="xs:boolean"
                    default ="true" minOccurs="0" maxOccurs="1" />
        <xs:element name ="MaxNumberOfViolations" default ="-1"
                    type ="xs:int" minOccurs ="0" maxOccurs ="1" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The **FunctionalDependencyProfileRequestType** complex type is an extension of the **TableProfileRequestType** complex type. The following table specifies additional information about the elements in the **FunctionalDependencyProfileRequestType** complex type.

| Element | Type definition | Description |
|---|---|---|
| DeterminantColumns | ColumnParameterCollectionType | A complex type that specifies a collection of determinant columns, which are the columns that are assessed for their ability to determine the dependent columns. |
| DependentColumn | ColumnParameterType | A complex type that specifies the column for which the dependency on the determinant columns is computed. |
| ThresholdSetting | StrengthThresholdSettingType | An enumeration value that specifies what type of threshold to use to identify a key candidate. |
| FDStrengthThreshold | xs:double | A double value that specifies the minimum functional dependency threshold that is required to consider a dependency as existing. This value is used only if the value of the **ThresholdSetting** element is set to "Specified".<83> |
| VerifyOutputInFastMode | xs:boolean | A Boolean value that specifies whether the output verification step is performed in fast mode. TRUE specifies that output verification is performed in fast mode. FALSE specifies that output verification is not performed in fast mode.<84> |
| MaxNumberOfViolations | xs:int | An integer value that specifies the maximum number of candidate key violations to report in the output. This option is disabled when the **KeyStrengthThreshold** element is set to the value "Exact".<85> |

### 2.7.1.7.1.1.2.2.2.8   InclusionProfileRequest Element

The **InclusionProfileRequest** element is used to specify parameters, properties, and settings that are used to request value inclusion profiles. A value inclusion profile, an option of the Data Profiling Task, computes the overlap in the values between two columns or sets of columns. Thus, the value inclusion profile can determine whether a column or set of columns is appropriate to serve as a foreign key between the selected tables.

The **InclusionProfileRequest** element has the XSD **substitutionGroup** attribute set to the value "ProfileRequest"; therefore, the **InclusionProfileRequest** element is one of the elements that can be substituted for the **ProfileRequest** element.

The following is the XSD of the **InclusionProfileRequest** element declaration.

```
<xs:element name="InclusionProfileRequest"
            type="InclusionProfileRequestType"
            substitutionGroup="ProfileRequest" />
```

### 2.7.1.7.1.1.2.2.2.8.1  InclusionProfileRequestType

The **InclusionProfileRequestType** complex type is the type of the **InclusionProfileRequest**
element.

The following is the XSD of the **InclusionProfileRequestType** complex type.

```
<xs:complexType name="InclusionProfileRequestType" final="#all">
  <xs:complexContent>
    <xs:extension base="ProfileRequestType">
      <xs:sequence>
        <xs:element name="SubsetDataSourceID" type="xs:string"
                    minOccurs="0" maxOccurs="1" />
        <xs:element name="SubsetTable" type="TableQNameType"
                    minOccurs="1" maxOccurs="1" />
        <xs:element name="SubsetColumns"
                    type="ColumnParameterCollectionType"
                    minOccurs="1" maxOccurs="1" />
        <xs:element name="SupersetDataSourceID" type="xs:string"
                    minOccurs="0" maxOccurs="1" />
        <xs:element name="SupersetTable" type="TableQNameType"
                    minOccurs="1" maxOccurs="1" />
        <xs:element name="SupersetColumns"
                    type="ColumnParameterCollectionType"
                    minOccurs="1" maxOccurs="1" />
        <xs:element name="InclusionThresholdSetting"
                    type="StrengthThresholdSettingType"
                    minOccurs="1" maxOccurs="1" />
        <xs:element name="InclusionStrengthThreshold" type="xs:double"
                    minOccurs="0" maxOccurs="1" />
        <xs:element name="SupersetColumnsKeyThresholdSetting"
                    type="StrengthThresholdSettingType"
                    minOccurs="1" maxOccurs="1" />
        <xs:element name="SupersetColumnsKeyThreshold" type="xs:double"
                    minOccurs="0" maxOccurs="1" />
        <xs:element name="VerifyOutputInFastMode" type="xs:boolean"
                    default="true" minOccurs="0" maxOccurs="1" />
        <xs:element name="MaxNumberOfViolations" default="-1"
                    type="xs:int" minOccurs ="0" maxOccurs="1" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The **InclusionProfileRequestType** complex type is an extension of the base type, the
ProfileRequestType abstract type. The following table specifies additional information about the
elements in the **InclusionProfileRequestType** type.

| Element | Type definition | Description |
| --- | --- | --- |
| SubsetDataSourceID | xs:string | A string value that contains an identifier for the subset data source. |
| SubsetTable | TableQNameType | A complex type that specifies information to reference the table that is used as the subset table. |
| SubsetColumns | ColumnParameterCollectionType | A complex type that specifies the columns that constitute the subset for |

| Element | Type definition | Description |
|---|---|---|
| | | which determination of the value inclusion profile is computed. |
| SupersetDataSourceID | xs:string | A string value that contains an identifier for the superset data source.<86> |
| SupersetTable | TableQNameType | A complex type that specifies information to reference the table that is used as the superset table. |
| SupersetColumns | ColumnParameterCollectionType | A complex type that specifies the columns that constitute the superset for which determination of the value inclusion profile is computed. |
| InclusionThresholdSetting | StrengthThresholdSettingType | An enumeration value that specifies what type of threshold to use in the value inclusion profile. |
| InclusionStrengthThreshold | xs:double | A double value that specifies the minimum inclusion strength that is required to report the inclusion strength. This value is used only if the value of the **InclusionThresholdSetting** element is set to "Specified".<87> |
| SupersetColumnsKeyThresholdSetting | StrengthThresholdSettingType | An enumeration value that specifies what type of threshold to use in the value inclusion profile for a key. |
| SupersetColumnsKeyThreshold | xs:double | A double value that specifies the minimum inclusion strength that is required to report the inclusion strength of the suitability of the column as a key. This value is used only if the value of the **SupersetColumnsKeyThresholdSetting** element is set to "Specified".<88> |
| VerifyOutputInFastMode | xs:boolean | A Boolean value that specifies whether the output verification step is performed in fast mode. TRUE specifies that output verification is performed in fast mode. FALSE specifies that output verification is not performed in fast mode.<89> |
| MaxNumberOfViolations | xs:int | An integer value that specifies the maximum number of violations to report in the output. This option is disabled when the value of the **InclusionThresholdSetting** element is set to "Exact".<90> |

### 2.7.1.7.1.1.2.2.3   Abstract Base Types Used by Data Profiling Task

The following abstract types are used as a base for extension or restriction for instance types that are used by the Data Profiling Task.

▪   ColumnProfileRequestType

- TableProfileRequestType

#### 2.7.1.7.1.1.2.2.3.1   ColumnProfileRequestType

The **ColumnProfileRequestType** complex type is the base type from which many types that request a profile of a column are extended.

The following is the XSD of the **ColumnProfileRequestType** complex type.

```
<xs:complexType name="ColumnProfileRequestType" abstract="true">
  <xs:complexContent>
    <xs:extension base="TableProfileRequestType">
      <xs:sequence>
        <xs:element name="Column" type="ColumnParameterType"
                    minOccurs="1" maxOccurs="1" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The following table specifies additional information about the elements in the **ColumnProfileRequestType** complex type.

| Element | Type definition | Description |
|---------|-----------------|-------------|
| Column | ColumnParameterType | A complex type that specifies information that is sufficient to identify a column at a data source for which a profile is being requested. |

#### 2.7.1.7.1.1.2.2.3.1   TableProfileRequestType

The **TableProfileRequestType** complex type is the base type from which many types that request a profile of a table or of table elements are extended.

The following is the XSD of the **TableProfileRequestType** complex type.

```
<xs:complexType name="TableProfileRequestType" abstract="true">
  <xs:complexContent>
    <xs:extension base="ProfileRequestType">
      <xs:sequence>
        <xs:element name="DataSourceID" type="xs:string" minOccurs = "0"/>
        <xs:element name="Table" type="TableQNameType" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The following table specifies additional information about the elements in the **TableProfileRequestType** complex type.

| Element | Type definition | Description |
|---------|-----------------|-------------|
| DataSourceID | xs:string | A string value that contains an identifier for the data source for this table request. |
| Table | TableQNameType | A complex type that identifies a single table within a data source. |

### 2.7.1.7.1.1.2.2.4    Other Types Used by Data Profiling Task

The following types are referenced as element types by other complex types within the Data Profiling Task.

▪ TableQNameType

▪ ColumnParameterCollectionType

▪ ColumnParameterType

▪ StringCompareOptionsEnum

▪ StrengthThresholdSettingType

### 2.7.1.7.1.1.2.2.4.1   TableQNameType

The **TableQNameType** complex type is used to specify an existing table in a database.

The following is the XSD of the **TableQNameType** complex type.

```
<xs:complexType name="TableQNameType">
  <xs:attribute name="Database" type="xs:string" use="optional" />
  <xs:attribute name="Schema" type="xs:string" use="optional" />
  <xs:attribute name="Table" type="xs:string" use="optional" />
</xs:complexType>
```

The **TableQNameType** complex type MUST NOT contain elements.

The following table specifies additional information for the attributes of the **TableQNameType** complex type.

| Attribute | Description |
|-----------|-------------|
| Database | A string value that specifies the name of the database in which the table resides. |
| Schema | A string value that specifies the name of the schema within the database. |
| Table | A string value that specifies the name of the table within the schema. |

### 2.7.1.7.1.1.2.2.4.2   ColumnParameterCollectionType

The **ColumnParameterCollectionType** complex type specifies a collection of columns that are considered as key column candidates.

The following is the XSD of the **ColumnParameterCollectionType** complex type.

```
<xs:complexType name="ColumnParameterCollectionType">
  <xs:sequence>
    <xs:element name="Column" type="ColumnParameterType"
                minOccurs="0" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
```

The following table specifies additional information about the elements in the **ColumnParameterCollectionType** complex type.

| Element | Type definition | Description |
|---------|-----------------|-------------|
| Column | ColumnParameterType | A complex type that specifies a single existing column by pointing to the column at its external location. |

#### 2.7.1.7.1.1.2.2.4.2.1 ColumnParameterType

The **ColumnParameterType** complex type specifies a collection of columns that are considered as key column candidates.

The following is the XSD of the **ColumnParameterType** complex type.

```
<xs:complexType name="ColumnParameterType">
  <xs:attribute name="IsWildCard" type="xs:boolean" use ="required" />
  <xs:attribute name="ColumnName" type="xs:string" use="optional" />
  <xs:attribute name="StringCompareOptions" type="DTS:StringCompareOptionsEnum"
                use="optional" />
</xs:complexType>
```

The following table specifies additional information about the elements in the **ColumnParameterType** complex type.

| Element | Type definition | Description |
|---------|-----------------|-------------|
| IsWildCard | xs:boolean | This Boolean value specifies whether the wildcard character (*) has been specified to select all columns  TRUE specifies that the wildcard character has been specified; therefore, all columns in the table are selected.  FALSE specifies that the wildcard character has not been specified and that specific individual columns are selected. |
| ColumnName | xs:string | A string value that specifies the name of a column. |
| StringCompareOptions | StringCompareOptionsEnum | An enumeration value that specifies the options to be used in string comparison. |

#### 2.7.1.7.1.1.2.2.4.2.1.1 StringCompareOptionsEnum

The **StringCompareOptionsEnum** simple type contains the enumeration values for the **StringCompareOptions** element. **StringCompareOptionsEnum** is a bitmask, but it is a bitmask in which not all combinations are permitted. The XSD enumeration enumerates only the allowed value combinations.

The following is the XSD for the **StringCompareOptionsEnum** simple type.

```
<xs:simpleType name="StringCompareOptionsEnum">
  <xs:restriction base="xs:int">
    <xs:enumeration value="0"/>
    <xs:enumeration value="1024"/>
    <xs:enumeration value="1025"/>
    <xs:enumeration value="1026"/>
    <xs:enumeration value="1027"/>
    <xs:enumeration value="1032"/>
    <xs:enumeration value="1033"/>
    <xs:enumeration value="1034"/>
    <xs:enumeration value="1035"/>
    <xs:enumeration value="1040"/>
    <xs:enumeration value="1041"/>
    <xs:enumeration value="1042"/>
    <xs:enumeration value="1043"/>
```

```
            <xs:enumeration value="1048"/>
            <xs:enumeration value="1050"/>
            <xs:enumeration value="1051"/>
            <xs:enumeration value="32768"/>
        </xs:restriction>
    </xs:simpleType>
```

The following table specifies the enumeration values for the **StringCompareOptionsEnum** type.

| Enumeration value | Description |
|---|---|
| 0 | Default. |
| 1024 | Dictionary sort. |
| 1025 | Dictionary sort, ignore case. |
| 1026 | Dictionary sort, ignore non-space characters. |
| 1027 | Dictionary sort, ignore case, and ignore non-space characters. |
| 1032 | Dictionary sort, ignore kana type. |
| 1033 | Dictionary sort, ignore case, and ignore kana type. |
| 1034 | Dictionary sort, ignore non-space characters, and ignore kana type. |
| 1035 | Dictionary sort, ignore case, ignore non-space characters, and ignore kana type. |
| 1040 | Dictionary sort, ignore width. |
| 1041 | Dictionary sort, ignore case, and ignore width. |
| 1042 | Dictionary sort, ignore non-space characters, and ignore width. |
| 1043 | Dictionary sort, ignore case, ignore non-space characters, and ignore width. |
| 1048 | Dictionary sort, ignore kana type, and ignore width. |
| 1050 | Dictionary sort, ignore non-space characters, ignore kana type, and ignore width. |
| 1051 | Dictionary sort, ignore case, ignore non-space characters, ignore kana type, and ignore width. |
| 32768 | Binary sort. |

#### 2.7.1.7.1.1.2.2.4.3   StrengthThresholdSettingType

The **StrengthThresholdSettingType** type specifies the enumeration values for the **ThresholdSetting** element.

The following is the XSD of the **StrengthThresholdSettingType** type.

```
    <xs:simpleType name="StrengthThresholdSettingType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="None" />
            <xs:enumeration value="Specified" />
            <xs:enumeration value="Exact" />
        </xs:restriction>
    </xs:simpleType>
```

The following table specifies the enumeration values for the **StrengthThresholdSettingType** type.

| Enumeration value | Description |
|---|---|
| None | No threshold is specified. The key strength is reported, regardless of its value. |
| Specified | A threshold is specified as the value of the **KeyStrengthThreshold** element. The key strength is reported only if it is greater than the threshold. |
| Exact | No threshold is specified. The key strength is reported only if the selected columns are an exact key. |

### 2.7.1.7.1.1.3 DataProfileOutputType

The **DataProfileOutputType** complex type contains the computed profile results.<91>

The following is the XSD of the **DataProfileOutputType** complex type.

```
<xs:complexType name="DataProfileOutputType">
  <xs:sequence>
    <xs:element name="Profiles" type="ProfilesType" minOccurs="1" maxOccurs="1" />
  </xs:sequence>
</xs:complexType>
```

The following table specifies additional information about the elements and types for the **DataProfileOutputType** complex type.

| Element | Type definition | Description |
|---|---|---|
| Profiles | ProfilesType | A complex type that MAY contain an output profile.<92> |

#### 2.7.1.7.1.1.3.1    ProfilesType

The **ProfilesType** complex type contains the computed profile results. The **ProfilesType** complex type MUST be empty.

The following is the XSD of the **ProfilesType** complex type.

```
<xs:complexType name="ProfilesType">
  <xs:sequence/>
</xs:complexType>
```

### 2.7.1.8 Exec80PackageTaskData Element

*Applies to DTSX2 schema for version 2012/01 (DTSX2 2012/01)*

The **Exec80PackageTaskData** element is used to specify parameters, properties, and settings that are specific to the execution of a Data Transformation Services (DTS) 2000 Package as a task. The **Exec80PackageTaskData** element resides in the Exec80PackageTask namespace. The ExecutableObjectDataType type refers to this element within an XSD **choice** element, using the XSD **ref** attribute, in the following XSD fragment.

```
<xs:element ref="Exec80PackageTask:Exec80PackageTaskData"/>
```

### 2.7.1.8.1 Exec80PackageTask Namespace

*Applies to DTSX2 schema for version 2012/01 (DTSX2 2012/01)*

The **Exec80PackageTask** namespace specifies the elements and types that are used by the Execute DTS 2000 Package Task.

The following is the XSD declaration for the **Exec80PackageTask** namespace.

```
xmlns:Exec80PackageTask=
"www.microsoft.com/sqlserver/dts/tasks/exec80packagetask"
```

#### 2.7.1.8.1.1    Exec80PackageTaskData Element

*Applies to DTSX2 schema for version 2012/01 (DTSX2 2012/01)*

The following is the XSD for the element declaration for the **Exec80PackageTaskData** element in the Exec80PackageTask namespace.

```
<xs:element name="Exec80PackageTaskData" type="Exec80PackageTask:Exec80PackageTaskType"/>
```

##### 2.7.1.8.1.1.1 Exec80PackageTaskType

*Applies to DTSX2 schema for version 2012/01 (DTSX2 2012/01)*

The **Exec80PackageTaskType** complex type holds parameters, properties, and settings for an Execute DTS 2000 Package Task.

The following is the XSD for the **Exec80PackageTaskType** complex type.

```
<xs:complexType name="Exec80PackageTaskType">
  <xs:sequence>
    <xs:element name="DTS2000Package" type="xs:base64Binary"
                minOccurs="0" form="unqualified"/>
  </xs:sequence>
  <xs:attribute name="Location"
                type="Exec80PackageTask:Exec80PackageTaskLocationEnum"/>
  <xs:attribute name="PackageName" type="xs:string"/>
  <xs:attribute name="PackageID" type="xs:string"/>
  <xs:attribute name="PackageVersionGUID" type="xs:string"/>
  <xs:attribute name="SQLServer" type="xs:string"/>
  <xs:attribute name="SQLUsername" type="xs:string"/>
  <xs:attribute name="WindowsAuthentication">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="True"/>
        <xs:enumeration value="False"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="Filename" type="xs:string"/>
  <xs:attribute name="InnerGlobalVariables" type="xs:string"/>
  <xs:attribute name="OuterGlobalVariables" type="xs:string"/>
</xs:complexType>
```

The following table specifies the elements of the **Exec80PackageTaskType** complex type.

| Element | Type definition | Description |
|---|---|---|
| DTS2000Package | xs:base64Binary | This element stores the content of the DTS 2000 Package in binary format. If this element is present, the DTS 2000 package is not retrieved from an external location. |

The following table specifies additional information for the attributes of the **Exec80PackageTaskType** complex type.

| Attribute | Description |
|---|---|
| Location | This enumeration value specifies the storage location for the DTS 2000 package file. |
| PackageName | This string specifies the name of the DTS 2000 package that is executed. |
| PackageID | This GUID value specifies the ID of the DTS 2000 package that is executed. |
| PackageVersionGUID | A GUID value that is unique to every version of the DTS 2000 package.<93> |
| SQLServer | This string value contains the name of the server instance.<94> |
| SQLUsername | This string value contains the database username.<95> |
| WindowsAuthentication | An enumeration value that specifies whether Windows Authentication is used. "True" specifies that Windows Authentication is used. "False" specifies that Windows Authentication is not used. |
| Filename | This string value specifies the name of the file. This value is used only if the **Location** attribute has a value of "StorageFile". |
| InnerGlobalVariables | A string value that contains a collection of variables that are the inner variables for the task. |
| OuterGlobalVariables | A string value that contains a collection of variables that are the outer variables for the task. |

#### 2.7.1.8.1.1.1.1    Exec80PackageTaskLocationEnum

*Applies to DTSX2 schema for version 2012/01 (DTSX2 2012/01)*

The **Exec80PackageTaskLocationEnum** type contains the enumeration values for the **Location** attribute.

The following is the XSD for the **Exec80PackageTaskLocationEnum** type.

```
<xs:simpleType name="Exec80PackageTaskLocationEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="SqlServer"/>
    <xs:enumeration value="EmbeddedInTask"/>
    <xs:enumeration value="StorageFile"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **Exec80PackageTaskLocationEnum** type.

| Enumeration value | Description |
|---|---|
| SqlServer | Specifies that the DTS 2000 Package file is stored on a database server.<96> |
| EmbeddedInTask | Specifies that the DTS 2000 Package file is embedded inline in the package file. |
| StorageFile | Specifies that the DTS 2000 Package file is stored in a file in the file system. |

### 2.7.1.9 ExecutePackageTaskObjectDataType

The **ExecutePackageTaskObjectDataType** complex type is used to specify parameters, properties, and settings that are specific to the Execute Package Task.

The following is the XSD for the **ExecutePackageTaskObjectDataType** complex type.

```
<xs:complexType name="ExecutePackageTaskObjectDataType">
  <xs:sequence>
    <xs:element name="ExecuteOutOfProcess" form="unqualified">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="True"/>
          <xs:enumeration value="False"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="PackageName" type="xs:string" form="unqualified"/>
     <xs:element name="UseProjectReference" type="DTS:BooleanStringCap"
                 minOccurs="0" form="unqualified"/>
    <xs:element name="PackagePassword" type="DTS:PasswordElementType"
                form="unqualified"/>
    <xs:element name="PackageID" type="DTS:uuid" form="unqualified"/>
    <xs:element name="VersionID" type="DTS:uuid" form="unqualified"/>
    <xs:element name="Connection" type="xs:string" form="unqualified"/>
  </xs:sequence>
</xs:complexType>
```

The following table specifies the elements of the **ExecutePackageTaskObjectDataType** complex type.

| Element | Type definition | Description |
|---|---|---|
| ExecuteOutOfProcess | Inline in the XSD in this section. | This enumeration value specifies whether the package is executed in a separate process. "True" specifies that the package is executed in a separate process. "False" specifies that the package is executed within the parent process. |
| PackageName | xs:string | A string value that specifies the name of the package that is run. |
| UseProjectReference | BooleanStringCap | A value of type **BooleanStringCap** that specifies whether the package to be executed is located in the same project as the current package. "True" specifies that the package to execute is located in the same project as the current package. "False" specifies that the package to execute is not located in the same project as the current package. |

| Element | Type definition | Description |
|---------|----------------|-------------|
| PackagePassword | PasswordElementType | A string value that specifies the password for use if the package is encrypted. |
| PackageID | uuid | A UUID value that specifies the identifier of the package that is executed. |
| VersionID | uuid | A UUID value that specifies the identifier of the version of the package to execute. |
| Connection | xs:string | A string value that specifies a reference to a connection manager by its DTSID property value or by its **ObjectName** property value. |

### 2.7.1.10    ExecuteProcessDataObjectDataType

The **ExecuteProcessDataObjectDataType** complex type is used to specify parameters, properties, and settings that are specific to the Execute Process Task.

The following is the XSD for the **ExecuteProcessDataObjectDataType** complex type.

```
<xs:complexType name="ExecuteProcessDataObjectDataType">
  <xs:sequence>
  </xs:sequence>
  <xs:attribute name="Executable" type="xs:string" form="unqualified"
               use="optional" default=""/>
  <xs:attribute name="RequireFullFileName" form="unqualified"
               use="optional" type="DTS:BooleanStringCap" default="True"/>
  <xs:attribute name="Arguments" type="xs:string" form="unqualified"
               use="optional" default=""/>
  <xs:attribute name="WorkingDirectory" type="xs:string"
               use="optional" default="" form="unqualified"/>
  <xs:attribute name="TimeOut" type="xs:int" form="unqualified"
               use="optional" default="0"/>
  <xs:attribute name="FailTaskIfReturnCodeIsNotSuccessValue"
               use="optional" type="DTS:BooleanStringCap" default="True"
               form="unqualified">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="True"/>
        <xs:enumeration value="False"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="SuccessValue" type="xs:int" form="unqualified"
               use="optional" default="0"/>
  <xs:attribute name="TerminateAfterTimeout" form="unqualified"
               use="optional" default="True">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="True"/>
        <xs:enumeration value="False"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="StandardInputVariable" type="xs:string"
               use="optional" default="" form="unqualified"/>
  <xs:attribute name="StandardOutputVariable" type="xs:string"
               use="optional" default="" form="unqualified"/>
  <xs:attribute name="StandardErrorVariable" type="xs:string"
               use="optional" default="" form="unqualified"/>
  <xs:attribute name="WindowStyle" form="unqualified"
```

```
                    type="DTS:WindowStyleEnum"
                    use="optional" default="Normal"/>
        </xs:complexType>
```

The **ExecuteProcessDataObjectDataType** complex type MUST NOT contain elements.

The following table specifies the attributes of the **ExecuteProcessDataObjectDataType** complex type.

| Attribute | Description |
|---|---|
| Executable | A string value that specifies the name of the executable to execute. |
| RequireFullFileName | An enumeration value that specifies whether the task fails if the executable is not found on the specified path. |
| | "True" specifies that the full file name is required and the task fails if the executable is not found on the specified path. |
| | "False" specifies that the full file name is not required and that the task can still execute if the executable is found in any location that is available to the process. |
| Arguments | A string value that specifies a list of arguments that is passed to the executable. |
| WorkingDirectory | A string value that specifies the working directory for the executable. |
| TimeOut | An integer value that specifies the timeout, in seconds, for the executable. |
| FailTaskIfReturnCodeIsNotSuccessValue | A value of type **DTS:BooleanStringCap** that specifies whether the task fails if the return code is not the same as that specified for the **SuccessValue** attribute. |
| | "True" specifies that the task fails if the return code is not the same as the value that is specified for the **SuccessValue** attribute. |
| | "False" specifies that the task does not fail if the return code is not the same as the value that is specified for the **SuccessValue** attribute. |
| SuccessValue | An integer value that specifies the value that is returned by the external process to indicate that the external process has completed successfully. |
| TerminateAfterTimeout | An enumeration value that specifies whether the task terminates after the timeout value is reached. |
| | "True" specifies that the task terminates after the timeout value is reached. |
| | "False" specifies that the task does not terminate after the timeout value is reached. |
| StandardInputVariable | A string value that specifies a variable that contains input information that is passed to the process. |
| StandardOutputVariable | A string value that specifies a variable where output information from the process is sent. |
| StandardErrorVariable | A string value that specifies a variable where error information from the process is sent. |
| WindowStyle | An enumeration value that specifies how the new window in which the task starts a process appears. |

### 2.7.1.10.1   WindowStyleEnum

The **WindowStyleEnum** type contains the enumeration values for specifying what the appearance is of the window that is created to run the process.

The following is the XSD for the **WindowStyleEnum** type.

```
<xs:simpleType name="WindowStyleEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Hidden"/>
    <xs:enumeration value="Maximized"/>
    <xs:enumeration value="Minimized"/>
    <xs:enumeration value="Normal"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **WindowStyleEnum** type.

| Enumeration value | Description |
|---|---|
| Hidden | Specifies that the window is hidden. |
| Maximized | Specifies that the window is maximized. |
| Minimized | Specifies that the window is minimized. |
| Normal | Specifies that the window is normal size. |

### 2.7.1.11      SqlTaskData Element

The **SqlTaskData** element is used to specify parameters, properties, and settings that are specific to the execution of an Execute SQL Task. This element resides in the SQLTask namespace. The ExecutableObjectDataType type refers to this element within an XSD **choice** element by using the XSD **ref** attribute, as shown in the following XSD fragment.

```
<xs:element ref="SQLTask:SqlTaskData"/>
```

### 2.7.1.11.1      SQLTask Namespace

The **SQLTask** namespace specifies the elements and types that are used by the Execute SQL Task.

The following is the XSD declaration for the **SQLTask** namespace.

```
xmlns:SQLTask="www.microsoft.com/sqlserver/dts/tasks/sqltask"
```

### 2.7.1.11.1.1  SqlTaskData Element

The **SqlTaskData** element is used to specify parameters, properties, and settings that are specific to the execution of an Execute SQL Task. The following is the XSD for the element declaration for the **SqlTaskData** element in the SQLTask namespace.

```
<xs:element name="SqlTaskData" type="SQLTask:SqlTaskDataType"/>
```

### 2.7.1.11.1.1.1 SqlTaskDataType

The **SqlTaskDataType** complex type holds parameters, properties, and settings for the Execute SQL Task.

The following is the XSD for the **SqlTaskDataType** complex type.

```
<xs:complexType name="SqlTaskDataType">
  <xs:sequence>
    <xs:element name="SelectedDatabases"
                type="SQLTask:SelectedDatabasesType"
                minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="OperatorNotifyList"
                type="SQLTask:OperatorNotifyListType"
                minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="ParameterBinding"
                type="SQLTask:SqlTaskParameterBindingType"
                minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="ResultBinding"
                type="SQLTask:SqlTaskResultBindingType"
                minOccurs="0" maxOccurs="1"/>
    <xs:element name="BackupDestinationList"
                type="SQLTask:BackupDestinationListType"
                minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attributeGroup ref="SQLTask:SqlTaskBaseAttributeGroup"/>
  <xs:attributeGroup ref="SQLTask:MaintenanceTaskBaseAttributeGroup"/>
  <xs:attributeGroup ref="SQLTask:BackupTaskAttributeGroup"/>
  <xs:attributeGroup ref="SQLTask:CheckIntegrityTaskAttributeGroup"/>
  <xs:attributeGroup ref="SQLTask:AgentJobTaskAttributeGroup"/>
  <xs:attributeGroup ref="SQLTask:HistoryCleanupTaskAttributeGroup"/>
  <xs:attributeGroup ref="SQLTask:MaintenanceFileCleanupTaskAttributeGroup"/>
  <xs:attributeGroup ref="SQLTask:NotifyOperatorTaskAttributeGroup"/>
  <xs:attributeGroup ref="SQLTask:RebuildIndexTaskAttributeGroup"/>
  <xs:attributeGroup ref="SQLTask:ReorganizeIndexTaskAttributeGroup"/>
  <xs:attributeGroup ref="SQLTask:ShrinkDBTaskAttributeGroup"/>
  <xs:attributeGroup ref="SQLTask:UpdateStatisticsTaskAttributeGroup"/>
  <xs:attribute name="RemoveOlderThan" type="xs:int"/>
  <xs:attribute name="TimeUnitsType" type="SQLTask:SqlTaskTimeUnitsTypeEnum"/>
  <xs:attribute name="DatabaseSelectionType"
                type="SQLTask:SqlTaskDatabaseSelectionTypeEnum"/>
</xs:complexType>
```

The following table provides additional information about the elements, types, and constraints in the **SqlTaskDataType** complex type.

| Element | Additional constraints | Type definition | Description |
|---------|------------------------|-----------------|-------------|
| SelectedDatabases | The **SelectedDatabases** element MUST NOT be used except with the following tasks: Backup Task, Check Integrity Task, Rebuild Index Task, Reorganize Index task, Shrink Database Task, and Update Statistics Task. | SelectedDatabasesType | A complex type value that specifies information about an existing database. |
| OperatorNotifyList | The **OperatorNotifyList** element MUST NOT be used except with the Notify Operator Task. | OperatorNotifyListType | A complex type value that specifies information about the operators to notify. |
| ParameterBinding | None. | SqlTaskParameterBindingType | A complex type |

| Element | Additional constraints | Type definition | Description |
|---|---|---|---|
| | | | value that specifies information about parameters that the Execute SQL Task uses as input or output. |
| ResultBinding | None. | SqlTaskResultBindingType | A complex type value that specifies information about the binding of the results of the execution of the Execute SQL Task. |
| BackupDestinationList | The **BackupDestinationList** element MUST NOT be used except with the Backup Task. | BackupDestinationListType | A complex type value that specifies information about the destinations for a Backup Task. |

The following table specifies additional information for the attributes and the attribute groups of the **SqlTaskDataType** complex type.

| Attribute or attribute group | Description |
|---|---|
| SqlTaskBaseAttributeGroup | An attribute group that specifies the attributes that are common to all SQL tasks. |
| MaintenanceTaskBaseAttributeGroup | An attribute group that specifies the attributes that are common to all maintenance tasks. |
| BackupTaskAttributeGroup | An attribute group that specifies the attributes for the Backup Task. |
| CheckIntegrityTaskAttributeGroup | An attribute group that specifies the attributes for the Check Integrity Task. |
| AgentJobTaskAttributeGroup | An attribute group that specifies the attributes for the Execute SQL Agent Job Task. |
| HistoryCleanupTaskAttributeGroup | An attribute group that specifies the attributes for the History Cleanup Task. |
| MaintenanceFileCleanupTaskAttributeGroup | An attribute group that specifies the attributes for the Maintenance File Cleanup Task. |
| NotifyOperatorTaskAttributeGroup | An attribute group that specifies the attributes for the Notify Operator Task. |
| RebuildIndexTaskAttributeGroup | An attribute group that specifies the attributes for the Rebuild Index Task. |
| ReorganizeIndexTaskAttributeGroup | An attribute group that specifies the attributes for the Reorganize Index Task. |
| ShrinkDBTaskAttributeGroup | An attribute group that specifies the attributes for the Shrink Database Task. |

| Attribute or attribute group | Description |
|---|---|
| UpdateStatisticsTaskAttributeGroup | An attribute group that specifies the attributes for the Update Statistics Task. |
| RemoveOlderThan | An integer value that specifies the number of units of time after which history is removed. The units for this integer value are specified by the **TimeUnitsType** attribute. |
| TimeUnitsType | An enumeration value that specifies the units of time for the **RemoveOlderThan** attribute. |
| DatabaseSelectionType | An enumeration value that specifies what types of databases are selected or whether a list of databases is specified. |

#### 2.7.1.11.1.1.1.1 SqlTaskDatabaseSelectionTypeEnum

The **SqlTaskDatabaseSelectionTypeEnum** type contains the enumeration values for the **DatabaseSelectionType** attribute.

The following is the XSD for the **SqlTaskDatabaseSelectionTypeEnum** type.

```
<xs:simpleType name="SqlTaskDatabaseSelectionTypeEnum">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="1"/>
    <xs:maxInclusive value="4"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **SqlTaskDatabaseSelectionTypeEnum** type.

| Enumeration value | Description |
|---|---|
| 1 | Specifies that all databases are selected. |
| 2 | Specifies that only system databases are selected. |
| 3 | Specifies that all user databases are selected. |
| 4 | Specifies that the selected databases are specified by **SelectedDatabases** elements. |

#### 2.7.1.11.1.1.1.2 SqlTaskTimeUnitsTypeEnum

The **SqlTaskTimeUnitsTypeEnum** type contains the enumeration values for the **TimeUnitsType** attribute.

The following is the XSD for the **SqlTaskTimeUnitsTypeEnum** type.

```
<xs:simpleType name="SqlTaskTimeUnitsTypeEnum">
  <xs:restriction base="xs:int">
    <xs:enumeration value="0"/>
    <xs:enumeration value="1"/>
    <xs:enumeration value="2"/>
    <xs:enumeration value="3"/>
    <xs:enumeration value="5"/>
```

```
        </xs:restriction>
    </xs:simpleType>
```

The following table specifies the enumeration values for the **SqlTaskTimeUnitsTypeEnum** type.

| Enumeration value | Description |
|---|---|
| 0 | The unit of time for the **RemoveOlderThan** attribute is days. |
| 1 | The unit of time for the **RemoveOlderThan** attribute is weeks. |
| 2 | The unit of time for the **RemoveOlderThan** attribute is months. |
| 3 | The unit of time for the **RemoveOlderThan** attribute is years. |
| 5 | The unit of time for the **RemoveOlderThan** attribute is hours. |

### 2.7.1.11.1.1.1.3    SelectedDatabasesType

The **SelectedDatabasesType** complex type contains information about the databases that are selected for an Execute SQL Task data operation.

The following is the XSD for the **SelectedDatabasesType** complex type.

```
<xs:complexType name="SelectedDatabasesType">
  <xs:sequence/>
  <xs:attribute name="DatabaseName" type="xs:string"/>
</xs:complexType>
```

The **SelectedDatabasesType** complex type MUST NOT contain any elements or values.

The following table specifies the attributes of the **SelectedDatabasesType** complex type.

| Attribute | Description |
|---|---|
| DatabaseName | A string value that contains the name of an existing database. |

### 2.7.1.11.1.1.1.4    OperatorNotifyListType

The **OperatorNotifyListType** complex type contains information about the databases that are selected for an Execute SQL Task data operation.

The following is the XSD for the **OperatorNotifyListType** complex type.

```
<xs:complexType name="OperatorNotifyListType">
  <xs:sequence/>
  <xs:attribute name="OperatorNotify" type="xs:string"/>
</xs:complexType>
```

The **OperatorNotifyListType** complex type MUST NOT contain any elements or values.

The following table specifies the attributes of the **OperatorNotifyListType** complex type.

| Attribute | Description |
|---|---|
| OperatorNotify | A string value that contains the name of an operator to notify. |

### 2.7.1.11.1.1.1.5    SqlTaskParameterBindingType

The **SqlTaskParameterBindingType** complex type specifies information about the parameters that are used to execute an SQL statement.

The following is the XSD for the **SqlTaskParameterBindingType** complex type.

```
<xs:complexType name="SqlTaskParameterBindingType">
  <xs:sequence/>
  <xs:attribute name="ParameterName" type="xs:string"/>
  <xs:attribute name="DtsVariableName" type="xs:string"/>
  <xs:attribute name="ParameterDirection" type="SQLTask:ParameterDirectionEnum"/>
  <xs:attribute name="DataType" type="DTS:DtsDataTypeEnum"/>
  <xs:attribute name="ParameterSize" type="xs:int"/>
</xs:complexType>
```

The **SqlTaskParameterBindingType** complex type MUST NOT contain any elements or values.

The following table specifies the attributes of the **SqlTaskParameterBindingType** complex type.

| Attribute | Description |
|---|---|
| ParameterName | A string value that specifies the name of the parameter. The format of this string value varies with the provider. |
| DtsVariableName | A string value that specifies the name of a DTS variable that provides an input value as the parameter value, receives an output value from the SQL execution, or receives a return value from the SQL execution. |
| ParameterDirection | An enumeration value that specifies the direction for the parameter. |
| DataType | An enumeration value that specifies the data type of the parameter. |
| ParameterSize | An integer value that specifies the size of the parameter. Whether a parameter needs a size specified depends on the type of the parameter and the requirement of the provider. |

### 2.7.1.11.1.1.1.5.1    ParameterDirectionEnum

The **ParameterDirectionEnum** type contains the enumeration values for the **ParameterDirection** attribute.

The following is the XSD for the **ParameterDirectionEnum** type.

```
<xs:simpleType name="ParameterDirectionEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Input"/>
    <xs:enumeration value="Output"/>
    <xs:enumeration value="ReturnValue"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **ParameterDirectionEnum** type.

| Enumeration value | Description |
|---|---|
| Input | Specifies that the parameter provides input to the Execute SQL Task. |
| Output | Specifies that the parameter receives output from the Execute SQL Task. |
| ReturnValue | Specifies that the parameter receives a return value from the Execute SQL Task. |

#### 2.7.1.11.1.1.1.6    BackupDestinationListType

The **BackupDestinationListType** complex type specifies a list of backup destinations.

The following is the XSD for the **BackupDestinationListType** complex type.

```
<xs:complexType name="BackupDestinationListType">
  <xs:sequence/>
  <xs:attribute name="BackupDestinationLocation" type="xs:string"/>
</xs:complexType>
```

The **BackupDestinationListType** complex type MUST NOT contain any elements or values.

The following table specifies the attributes of the **BackupDestinationListType** complex type.

| Attribute | Description |
|---|---|
| BackupDestinationLocation | A string value that specifies the destination location for the backup. |

#### 2.7.1.11.1.1.1.7    SqlTaskResultBindingType

The **SqlTaskResultBindingType** complex type specifies information about parameters that are used to execute an SQL statement.

The following is the XSD for the **SqlTaskResultBindingType** complex type.

```
<xs:complexType name="SqlTaskResultBindingType">
  <xs:sequence/>
  <xs:attribute name="ResultName" type="xs:string"/>
  <xs:attribute name="DtsVariableName" type="xs:string"/>
</xs:complexType>
```

The **SqlTaskResultBindingType** complex type MUST NOT contain any elements or values.

The following table specifies the attributes of the **SqlTaskResultBindingType** complex type.

| Attribute | Description |
|---|---|
| ResultName | A string value that specifies the name of the result. The format of this string value varies with the provider. |
| DtsVariableName | A string value that specifies the name of a DTS variable that receives the result. |

### 2.7.1.11.1.1.1.8 SqlTaskBaseAttributeGroup

The **SqlTaskBaseAttributeGroup** attribute group specifies the attributes that are used by the SqlTaskData element for all tasks that use the **SqlTaskData** element.

The following is the XSD for the **SqlTaskBaseAttributeGroup** attribute group.

```
<xs:attributeGroup name="SqlTaskBaseAttributeGroup">
  <xs:attribute name="Connection" type="DTS:uuid" use="optional" default=""/>
  <xs:attribute name="TimeOut" type="xs:int" use="optional" default="0"/>
  <xs:attribute name="IsStoredProc" use="optional" default="False">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="True"/>
        <xs:enumeration value="False"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="BypassPrepare" use="optional" default="True">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="True"/>
        <xs:enumeration value="False"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="SqlStmtSourceType" type="DTS:SourceTypeEnum"
              use="optional" default="DirectInput"/>
  <xs:attribute name="SqlStatementSource" type="xs:string"
              use="optional" default=""/>
  <xs:attribute name="CodePage" type="xs:int" use="optional"/>
  <xs:attribute name="ResultType" type="SQLTask:SqlResultTypeEnum"
              use="optional" default="ResultSetType_None"/>
</xs:attributeGroup>
```

The following table specifies additional information for the attributes of the **SqlTaskBaseAttributeGroup** attribute group.

| Attribute | Description |
|---|---|
| Connection | A string value that specifies a reference to a connection manager by its **DTSID** property value or by its **ObjectName** property value. |
| TimeOut | An integer that specifies the timeout interval, in seconds, for the task. |
| IsStoredProc | An enumeration value that specifies whether the Execute SQL Task executes a stored procedure. "True" specifies that the task executes a stored procedure. "False" specifies that the task does not execute a stored procedure. |
| BypassPrepare | An enumeration value that specifies whether the SQL statement is prepared before the SQL statement is executed. "True" specifies that the prepare step is bypassed and that the task is not prepared before it is executed. "False" specifies that the prepare step is not bypassed and that the task is prepared before it is executed. |
| SqlStmtSourceType | This enumeration value specifies the source of the SQL statement. |
| SqlStatementSource | This string value specifies the SQL statement. The string value can contain an SQL language statement, or it can contain identifying information for the source of the SQL statement. The content depends on the value of the **SqlStmtSourceType** attribute. |

| Attribute | Description |
|-----------|-------------|
| CodePage | This integer value specifies the code page for the task.<97> |
| ResultType | This enumeration value specifies the form of the output from the task. |

#### 2.7.1.11.1.1.1.8.1   SqlResultTypeEnum

The **SqlResultTypeEnum** type contains the enumeration values for the **SqlResultType** attribute.

The following is the XSD for the **SqlResultTypeEnum** type.

```
<xs:simpleType name="SqlResultTypeEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="ResultSetType_None"/>
    <xs:enumeration value="ResultSetType_Rowset"/>
    <xs:enumeration value="ResultSetType_SingleRow"/>
    <xs:enumeration value="ResultSetType_XML"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **SqlResultTypeEnum** type.

| Enumeration value | Description |
|-------------------|-------------|
| ResultSetType_None | Specifies that no result set is returned. |
| ResultSetType_Rowset | Specifies that the result set that is returned is a rowset. |
| ResultSetType_SingleRow | Specifies that the result set that is returned is a single row. |
| ResultSetType_XML | Specifies that the result set that is returned is XML. |

#### 2.7.1.11.1.1.1.9   MaintenanceTaskBaseAttributeGroup

The **MaintenanceTaskBaseAttributeGroup** attribute group specifies the attributes that are used by the SqlTaskData element by all maintenance plan tasks. The attributes that are specified in this section MUST NOT be used unless the task is one of the maintenance plan tasks.

The following is the XSD for the **MaintenanceTaskBaseAttributeGroup** attribute group.

```
<xs:attributeGroup name="MaintenanceTaskBaseAttributeGroup">
  <xs:attribute name="ServerVersion" type="xs:int"/>
  <xs:attribute name="ExtendedLogging">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="True"/>
        <xs:enumeration value="False"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="LocalConnectionForLogging" type="xs:string"/>
  <xs:attribute name="TaskName" type="xs:string"/>
  <xs:attribute name="IgnoreDatabasesInNotOnlineState">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="True"/>
```

```
            <xs:enumeration value="False"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:attributeGroup>
```

The following table specifies additional information for the attributes of the
**MaintenanceTaskBaseAttributeGroup** attribute group.

| Attribute | Description |
|---|---|
| ServerVersion | An integer value that specifies the server version. |
| ExtendedLogging | An enumeration value that specifies whether to include additional information about server operation at the end of a successful execution of a maintenance task in the log file.<br>"True" specifies that additional information is included in the log file.<br>"False" specifies that additional information is not included in the log file. |
| LocalConnectionForLogging | A string value that specifies a local connection manager that is used for logging. The string can reference the connection manager by pointing to the **DTSID** property of the connection manager or to the **ObjectName** property of the connection manager. |
| TaskName | A string value that specifies the name of the task. |
| IgnoreDatabasesInNotOnlineState | An enumeration value that specifies whether databases that are not in the online state are ignored.<br>"True" specifies that databases that are not in the online state are ignored.<br>"False" specifies that databases that are not in the online state are not ignored. |

### 2.7.1.11.1.1.1.10    BackupTaskAttributeGroup

The **BackupTaskAttributeGroup** attribute group specifies the attributes that are used by the
SqlTaskData element when the **ExecutableType** attribute of the containing Executable element is
equal to one of the following values:

▪   Microsoft.SqlServer.Management.DatabaseMaintenance.DbMaintenanceBackupTask,
    Microsoft.SqlServer.MaintenancePlanTasks, Version=11.0.0.0, Culture=neutral,
    PublicKeyToken=89845dcd8080cc91 (for DTSX2 2012/01)

▪   Microsoft.DbMaintenanceBackupTask (for DTSX2 2014/01)

The attributes that are specified in this section MUST NOT be used unless the **ExecutableType**
attribute is equal to one of these values.

The following is the XSD for the **BackupTaskAttributeGroup** attribute group.

```
    <xs:attributeGroup name="BackupTaskAttributeGroup">
      <xs:attribute name="BackupAction" type="SQLTask:BackupActionEnum"/>
      <xs:attribute name="BackupIsIncremental">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="True"/>
            <xs:enumeration value="False"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
```

```
                <xs:attribute name="BackupFileGroupsFiles" type="xs:string"/>
                <xs:attribute name="BackupDeviceType"
                              type="SQLTask:BackupDeviceTypeEnum"/>
                <xs:attribute name="BackupPhisycalDestinationType"
                              type="SQLTask:BackupDeviceTypeEnum"/>
                <xs:attribute name="BackupDestinationType"
                              type="SQLTask:DestinationTypeEnum"/>
                <xs:attribute name="BackupDestinationAutoFolderPath"
                              type="xs:string"/>
                <xs:attribute name="BackupActionForExistingBackups"
                              type="SQLTask:BackupActionForExistingBackupsEnum"/>
                <xs:attribute name="BackupCreateSubFolder">
                  <xs:simpleType>
                    <xs:restriction base="xs:string">
                      <xs:enumeration value="True"/>
                      <xs:enumeration value="False"/>
                    </xs:restriction>
                  </xs:simpleType>
                </xs:attribute>
                <xs:attribute name="BackupFileExtension" type="xs:string"/>
                <xs:attribute name="BackupVerifyIntegrity">
                  <xs:simpleType>
                    <xs:restriction base="xs:string">
                      <xs:enumeration value="True"/>
                      <xs:enumeration value="False"/>
                    </xs:restriction>
                  </xs:simpleType>
                </xs:attribute>
                <xs:attribute name="ExpireDate" type="xs:dateTime"/>
                <xs:attribute name="RetainDays" type="xs:int"/>
                <xs:attribute name="InDays">
                  <xs:simpleType>
                    <xs:restriction base="xs:string">
                      <xs:enumeration value="True"/>
                      <xs:enumeration value="False"/>
                    </xs:restriction>
                  </xs:simpleType>
                </xs:attribute>
                <xs:attribute name="UseExpiration">
                  <xs:simpleType>
                    <xs:restriction base="xs:string">
                      <xs:enumeration value="True"/>
                      <xs:enumeration value="False"/>
                    </xs:restriction>
                  </xs:simpleType>
                </xs:attribute>
                <xs:attribute name="BackupCompressionAction"
                              type="SQLTask:BackupCompressionActionEnum"/>
                <xs:attribute name="BackupTailLog">
                  <xs:simpleType>
                    <xs:restriction base="xs:string">
                      <xs:enumeration value="True"/>
                      <xs:enumeration value="False"/>
                    </xs:restriction>
                  </xs:simpleType>
                </xs:attribute>
            </xs:attributeGroup>
```

The following table specifies additional information for the attributes of the
**BackupTaskAttributeGroup** attribute group.

| Attribute | Description |
|---|---|
| BackupAction | An enumeration value that specifies the action to be taken by the Backup Task. |
| BackupIsIncremental | An enumeration value that specifies whether the backup is incremental. |

| Attribute | Description |
|---|---|
| | "True" specifies that the backup is incremental.<br>"False" specifies that the backup is not incremental. |
| BackupFileGroupsFiles | A string value that specifies the names of files or file groups that are to be backed up. Multiple names are specified as a comma-separated list. |
| BackupDeviceType | An enumeration value that specifies the type of device that the backup is output to. |
| BackupPhysicalDestinationType | An enumeration value that specifies the type of device that the backup is output to. If the value is 1 (tape), then no rewind or unload of the backup tape is issued after the completion of the backup. |
| BackupDestinationType | An enumeration value that specifies the type of the destination for the backup. |
| BackupDestinationAutoFolderPath | A string value that specifies a folder path where further folders are created for the backup output. |
| BackupActionForExistingBackups | An enumeration value that specifies the action to take if an existing backup exists at the destination. |
| BackupCreateSubFolder | An enumeration value that specifies whether to create subfolders for the backups.<br>"True" specifies that subfolders are created.<br>"False" specifies that subfolders are not created. |
| BackupFileExtension | A string value that specifies the file extension for files that are created by the backup. |
| BackupVerifyIntegrity | An enumeration value that specifies whether the task verifies the integrity of the backup after the backup file is created.<br>"True" specifies that the task verifies the integrity of the backup file that is created.<br>"False" specifies that the task does not verify the integrity of the backup file that is created. |
| ExpireDate | A date/time value that specifies the expiration date for the backup. |
| RetainDays | An integer value that specifies a number of days until the backup expires. |
| InDays | An enumeration value that specifies whether to use the **ExpireDate** attribute value or the **RetainDays** attribute value for the expiration of the backup.<br>"True" specifies that the **RetainDays** attribute value is used to determine the backup expiration.<br>"False" specifies that the **ExpireDate** attribute value is used to determine the backup expiration. |
| UseExpiration | An enumeration value that specifies whether the backup expires.<br>"True" specifies that the backup expires.<br>"False" specifies that the backup does not expire. |
| BackupCompressionAction | An enumeration value that specifies whether the backup is compressed, the backup is not compressed, or the default compression setting is used. |
| BackupTailLog | An enumeration value that specifies whether the Backup Task backs up the tail of the log at the end of the task.<br>"True" specifies that the tail of the log is backed up at the end of the backup task.<br>"False" specifies that the tail of the log is not backed up at the end of the |

| Attribute | Description |
|---|---|
| | backup task. |

#### 2.7.1.11.1.1.1.10.1 BackupActionEnum

The **BackupActionEnum** type contains the enumeration values for the **BackupAction** attribute.

The following is the XSD for the **BackupActionEnum** type.

```
<xs:simpleType name="BackupActionEnum">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="2"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **BackupActionEnum** type.

| Enumeration value | Description |
|---|---|
| 0 | Back up a database. |
| 1 | Back up files. |
| 2 | Back up a transaction log. |

#### 2.7.1.11.1.1.1.10.2 BackupDeviceTypeEnum

The **BackupDeviceTypeEnum** type contains the enumeration values for the **BackupDeviceType** attribute.

The following is the XSD for the **BackupDeviceTypeEnum** type.

```
<xs:simpleType name="BackupDeviceTypeEnum">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="4"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **BackupDeviceTypeEnum** type.

| Enumeration value | Description |
|---|---|
| 0 | Specifies that the backup device is a logical device. |
| 1 | Specifies that the backup device is a tape. |
| 2 | Specifies that the backup device is a file. |
| 3 | Specifies that the backup device is a pipe. |
| 4 | Specifies that the backup device is a virtual device. |

### 2.7.1.11.1.1.1.10.3 DestinationTypeEnum

The **DestinationTypeEnum** type contains the enumeration values for the **DestinationType** attribute.

The following is the XSD for the **DestinationTypeEnum** type.

```
<xs:simpleType name="DestinationTypeEnum">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="1"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **DestinationTypeEnum** type.

| Enumeration value | Description |
|---|---|
| 0 | Specifies that the destination is automatically determined by the task. |
| 1 | Specifies that the destination is manually set by the user. |

### 2.7.1.11.1.1.1.10.4 BackupActionForExistingBackupsEnum

The **BackupActionForExistingBackupsEnum** type contains the enumeration values for the **BackupActionForExistingBackups** attribute.

The following is the XSD for the **BackupActionForExistingBackupsEnum** type.

```
<xs:simpleType name="BackupActionForExistingBackupsEnum">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="1"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **BackupActionForExistingBackupsEnum** type.

| Enumeration value | Description |
|---|---|
| 0 | Specifies that existing backups are appended to. |

| Enumeration value | Description |
|---|---|
| 1 | Specifies that existing backups are overwritten. |

### 2.7.1.11.1.1.1.10.5 BackupCompressionActionEnum

The **BackupCompressionActionEnum** type contains the enumeration values for the **CompressionAction** attribute.

The following is the XSD for the **BackupCompressionActionEnum** type.

```
<xs:simpleType name="BackupCompressionActionEnum">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="2"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **BackupCompressionActionEnum** type.

| Enumeration value | Description |
|---|---|
| 0 | Specifies that the default setting for compression is used. |
| 1 | Specifies that backups are compressed. |
| 2 | Specifies that backups are not compressed. |

### 2.7.1.11.1.1.1.11    CheckIntegrityTaskAttributeGroup

The **CheckIntegrityTaskAttributeGroup** attribute group specifies the attributes that are used by the SqlTaskData element when the **ExecutableType** attribute of the containing Executable element is equal to one of the following values:

- Microsoft.SqlServer.Management.DatabaseMaintenance.DbMaintenanceCheckIntegrityTask, Microsoft.SqlServer.MaintenancePlanTasks, Version=11.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91 (for DTSX2 2012/01)

- Microsoft.DbMaintenanceCheckIntegrityTask (for DTSX2 2014/01)

The attributes that are specified in this section MUST NOT be used unless the **ExecutableType** attribute is equal to one of these values.

The following is the XSD for the **CheckIntegrityTaskAttributeGroup** attribute group.

```
<xs:attributeGroup name="CheckIntegrityTaskAttributeGroup">
  <xs:attribute name="IncludeIndexes">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="True"/>
        <xs:enumeration value="False"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:attributeGroup>
```

The following table specifies additional information for the attributes of the **CheckIntegrityTaskAttributeGroup** attribute group.

| Attribute | Description |
|---|---|
| IncludeIndexes | An enumeration value that specifies whether to check the integrity of all indexes in addition to the table data pages.<br><br>"True" specifies that the integrity of all indexes is checked in addition to the table data pages.<br><br>"False" specifies that the integrity of only the table data pages is checked. |

### 2.7.1.11.1.1.1.12 ReorganizeIndexTaskAttributeGroup

The **ReorganizeIndexTaskAttributeGroup** attribute group specifies the attributes that are utilized by the SqlTaskData element when the **ExecutableType** attribute of the containing Executable element is equal to one of the following values:

- Microsoft.SqlServer.Management.DatabaseMaintenance.DbMaintenanceDefragmentIndexTask, Microsoft.SqlServer.MaintenancePlanTasks, Version=11.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91 (for DTSX2 2012/01)

- Microsoft.DbMaintenanceDefragmentIndexTask (for DTSX2 2014/01)

The attributes that are specified in this section MUST NOT be used unless the **ExecutableType** attribute is equal to one of these values.

The following is the XSD for the **ReorganizeIndexTaskAttributeGroup** attribute group.

```
<xs:attributeGroup name="ReorganizeIndexTaskAttributeGroup">
  <xs:attribute name="CompactLargeObjects">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="True"/>
        <xs:enumeration value="False"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:attributeGroup>
```

The following table specifies additional information for the attributes of the **ReorganizeIndexTaskAttributeGroup** attribute group.

| Attribute | Description |
|---|---|
| CompactLargeObjects | An enumeration value that specifies whether space for tables and views is deallocated when possible.<br><br>"True" specifies that space for tables and views is deallocated when possible.<br><br>"False" specifies that large objects are not compacted by the task. |

### 2.7.1.11.1.1.1.13 AgentJobTaskAttributeGroup

The **AgentJobTaskAttributeGroup** attribute group specifies the attributes that are used by the SqlTaskData element when the **ExecutableType** attribute of the containing Executable element is equal to one of the following values:

- Microsoft.SqlServer.Management.DatabaseMaintenance.DbMaintenanceExecuteAgentJobTask, Microsoft.SqlServer.MaintenancePlanTasks, Version=11.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91 (for DTSX2 2012/01)

- Microsoft.DbMaintenanceExecuteAgentJobTask (for DTSX2 2014/01)

The attributes that are specified in this section MUST NOT be used unless the **ExecutableType** attribute is equal to one of these values.

The following is the XSD for the **AgentJobTaskAttributeGroup** attribute group.

```
<xs:attributeGroup name="AgentJobTaskAttributeGroup">
  <xs:attribute name="AgentJobID" type="xs:string"/>
</xs:attributeGroup>
```

The following table specifies additional information for the attributes of the **AgentJobTaskAttributeGroup** attribute group.

| Attribute | Description |
|-----------|-------------|
| AgentJobID | A string value that specifies the job identifier for the Execute SQL Agent Job Task. |

#### 2.7.1.11.1.1.1.14    HistoryCleanupTaskAttributeGroup

The **HistoryCleanupTaskAttributeGroup** attribute group specifies the attributes that are used by the SqlTaskData element when the **ExecutableType** attribute of the containing Executable element is equal to one of the following values:

- Microsoft.SqlServer.Management.DatabaseMaintenance.DbMaintenanceHistoryCleanupTask, Microsoft.SqlServer.MaintenancePlanTasks, Version=11.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91 (for DTSX2 2012/01)

- Microsoft.DbMaintenanceHistoryCleanupTask (for DTSX2 2014/01)

The attributes that are specified in this section MUST NOT be used unless the **ExecutableType** attribute is equal to one of these values.

The following is the XSD for the **HistoryCleanupTaskAttributeGroup** attribute group.

```
<xs:attributeGroup name="HistoryCleanupTaskAttributeGroup">
  <xs:attribute name="RemoveBackupRestoreHistory">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="True"/>
        <xs:enumeration value="False"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="RemoveAgentHistory">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="True"/>
        <xs:enumeration value="False"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="RemoveDbMaintHistory">
    <xs:simpleType>
      <xs:restriction base="xs:string">
```

```
                  <xs:enumeration value="True"/>
                  <xs:enumeration value="False"/>
               </xs:restriction>
            </xs:simpleType>
         </xs:attribute>
      </xs:attributeGroup>
```

The following table specifies additional information for the attributes of the
**HistoryCleanupTaskAttributeGroup** attribute group.

| Attribute | Description |
|---|---|
| RemoveBackupRestoreHistory | An enumeration value that specifies whether the History Cleanup Task removes the backup restore history.<br><br>"True" specifies that the History Cleanup Task removes the backup restore history.<br><br>"False" specifies that the History Cleanup Task does not remove the backup restore history. |
| RemoveAgentHistory | An enumeration value that specifies whether the History Cleanup Task removes the SQL agent history.<br><br>"True" specifies that the History Cleanup Task removes the SQL agent history.<br><br>"False" specifies that the History Cleanup Task does not remove the SQL agent history. |
| RemoveDBMaintHistory | An enumeration value that specifies whether the History Cleanup Task removes database maintenance history.<br><br>"True" specifies that the History Cleanup Task removes the database maintenance history.<br><br>"False" specifies that the History Cleanup Task does not remove the database maintenance history. |

### 2.7.1.11.1.1.1.15   MaintenanceFileCleanupTaskAttributeGroup

The **MaintenanceFileCleanupTaskAttributeGroup** attribute group specifies the attributes that are
used by the SqlTaskData element when the **ExecutableType** attribute of the containing Executable
element is equal to one of the following values:

▪   Microsoft.SqlServer.Management.DatabaseMaintenance.DbMaintenanceFileCleanupTask,
    Microsoft.SqlServer.MaintenancePlanTasks, Version=11.0.0.0, Culture=neutral,
    PublicKeyToken=89845dcd8080cc91 (for DTSX2 2012/01)

▪   Microsoft.DbMaintenanceFileCleanupTask (for DTSX2 2014/01)

The attributes that are specified in this section MUST NOT be used unless the **ExecutableType**
attribute is equal to one of these values.

The following is the XSD for the **MaintenanceFileCleanupTaskAttributeGroup** attribute group.

```
      <xs:attributeGroup name="MaintenanceFileCleanupTaskAttributeGroup">
         <xs:attribute name="FileTypeSelected" type="xs:int"/>
         <xs:attribute name="FilePath" type="xs:string"/>
         <xs:attribute name="FolderPath" type="xs:string"/>
         <xs:attribute name="CleanSubFolders">
            <xs:simpleType>
               <xs:restriction base="xs:string">
                  <xs:enumeration value="True"/>
                  <xs:enumeration value="False"/>
```

```
            </xs:restriction>
          </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="FileExtension" type="xs:string"/>
      <xs:attribute name="AgeBased">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="True"/>
            <xs:enumeration value="False"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="DeleteSpecificFile">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="True"/>
            <xs:enumeration value="False"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
  </xs:attributeGroup>
```

The following table specifies additional information for the attributes of the
**MaintenanceFileCleanupTaskAttributeGroup** attribute group.

| Attribute | Description |
|---|---|
| FileTypeSelected | An integer value that specifies the type of the files that are selected to be deleted. "0" specifies that backup files are selected for deletion. "1" specifies that maintenance plan text reports are selected for deletion. |
| FilePath | A string value that contains the path and the name of the file to be deleted. |
| FolderPath | A string value that contains the path and the name of the folder that contains the files to be deleted. |
| CleanSubFolders | An enumeration value that specifies whether the task removes files in first-level subfolders to the folder that is specified in the **FolderPath** attribute. "True" specifies that the task removes files in first-level subfolders to the folder that is specified in the **FolderPath** attribute. "False" specifies that the task does not remove files in first-level subfolders to the folder specified in the **FolderPath** attribute. |
| FileExtension | A string value that specifies the file extension for the files to be deleted. |
| AgeBased | An enumeration value that specifies whether the task deletes all files of the specified type that are older than the age that is specified in the **RemoveOlderThan** attribute. "True" specifies that the task removes all files of the specified type that are older than the specified age. "False" specifies that the task does not remove files of the specified type that are older than the specified age. |
| DeleteSpecificFile | An enumeration value that specifies whether the task deletes a specific file that is specified in the **FilePath** attribute. "True" specifies that the task deletes a specific file that is specified in the **FilePath** attribute. "False" specifies that the task does not delete a specific file that is specified in the **FilePath** attribute. |

### 2.7.1.11.1.1.1.15.1 SqlTaskFileTypeSelectedEnum

The **SqlTaskFileTypeSelectedEnum** type contains the enumeration values for the **FileTypeSelected** attribute.

The following is the XSD for the **SqlTaskFileTypeSelectedEnum** type.

```
<xs:simpleType name="SqlTaskFileTypeSelectedEnum">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="1"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **SqlTaskFileTypeSelectedEnum** type.

| Enumeration value | Description |
|---|---|
| 0 | Specifies that backup files are selected for removal. |
| 1 | Specifies that maintenance plan text reports are selected for removal. |

### 2.7.1.11.1.1.1.16    NotifyOperatorTaskAttributeGroup

The **NotifyOperatorTaskAttributeGroup** attribute groups specifies the attributes that are used by the SqlTaskData element when the **ExecutableType** attribute of the containing Executable element is equal to one of the following values:

- Microsoft.SqlServer.Management.DatabaseMaintenance.DbMaintenanceNotifyOperatorTask, Microsoft.SqlServer.MaintenancePlanTasks, Version=11.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91 (for DTSX2 2012/01)

- Microsoft.DbMaintenanceNotifyOperatorTask (for DTSX2 2014/01)

The attributes that are specified in this section MUST NOT be used unless the **ExecutableType** attribute is equal to one of these values.

The following is the XSD for the **NotifyOperatorTaskAttributeGroup** attribute group.

```
<xs:attributeGroup name="NotifyOperatorTaskAttributeGroup">
  <xs:attribute name="Message" type="xs:string"/>
  <xs:attribute name="Profile" type="xs:string"/>
  <xs:attribute name="Subject" type="xs:string"/>
</xs:attributeGroup>
```

The following table specifies additional information for the attributes of the **NotifyOperatorTaskAttributeGroup** attribute group.

| Attribute | Description |
|---|---|
| Message | A string value that contains the message text for the operator notification. |
| Profile | The name of the database mail profile to send to. |
| Subject | A string value that contains the subject line text for the operator notification. |

### 2.7.1.11.1.1.1.17    RebuildIndexTaskAttributeGroup

The **RebuildIndexTaskAttributeGroup** attribute group specifies the attributes that are used by the SqlTaskData element when the **ExecutableType** attribute of the containing Executable element is equal to one of the following values:

- Microsoft.SqlServer.Management.DatabaseMaintenance.DbMaintenanceReindexTask, Microsoft.SqlServer.MaintenancePlanTasks, Version=11.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91 (for DTSX2 2012/01)

- Microsoft.DbMaintenanceReindexTask (for DTSX2 2014/01)

The attributes that are specified in this section MUST NOT be used unless the **ExecutableType** attribute is equal to one of these values.

The following is the XSD for the **RebuildIndexTaskAttributeGroup** attribute group.

```
<xs:attributeGroup name="RebuildIndexTaskAttributeGroup">
  <xs:attribute name="UseOriginalAmount">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="True"/>
        <xs:enumeration value="False"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="Percentage" type="xs:int"/>
  <xs:attribute name="Sort">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="True"/>
        <xs:enumeration value="False"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="KeepOnline">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="True"/>
        <xs:enumeration value="False"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:attributeGroup>
```

The following table specifies additional information for the attributes of the **RebuildIndexTaskAttributeGroup** attribute group.

| Attribute | Description |
|---|---|
| UseOriginalAmount | An enumeration value that specifies that the indexes are rebuilt with the same amount of free space as was allocated when the index was created.<br>"True" specifies that indexes are rebuilt with the amount of free space that was allocated when the index was created.<br>"False" specifies that indexes are not rebuilt with the amount of free space that was allocated when the index was created. |
| Percentage | An integer value that specifies the percentage of free space to allocate in the rebuilt indexes. |
| Sort | An enumeration value that specifies whether sorts that are required during the rebuild index operation are performed in a temporary database. |

| Attribute | Description |
|---|---|
| | "True" specifies that any sorts that are required during the rebuild index operation are performed in a temporary database. |
| | "False" specifies that any sorts that are required during the rebuild index operation are not performed in a temporary database. |
| KeepOnline | An enumeration value that specifies whether the database is to be kept online during the rebuild index operation. |
| | "True" specifies that the index is to be kept online during the rebuild index operation. |
| | "False" specifies that the index is not to be kept online during the rebuild index operation. |

### 2.7.1.11.1.1.1.18    ShrinkDBTaskAttributeGroup

The **ShrinkDBTaskAttributeGroup** attribute group specifies the attributes that are used by the SqlTaskData element when the **ExecutableType** attribute of the containing Executable element is equal to one of the following values:

▪ Microsoft.SqlServer.Management.DatabaseMaintenance.DbMaintenanceShrinkTask, Microsoft.SqlServer.MaintenancePlanTasks, Version=11.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91 (for DTSX2 2012/01)

▪ Microsoft.DbMaintenanceShrinkTask (for DTSX2 2014/01)

The attributes that are specified in this section MUST NOT be used unless the **ExecutableType** attribute is equal to one of these values.

The following is the XSD for the **ShrinkDBTaskAttributeGroup** attribute group.

```
<xs:attributeGroup name="ShrinkDBTaskAttributeGroup">
  <xs:attribute name="DatabaseSizeLimit" type="xs:int"/>
  <xs:attribute name="DatabasePercentLimit" type="xs:int"/>
  <xs:attribute name="DatabaseReturnFreeSpace">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="True"/>
        <xs:enumeration value="False"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:attributeGroup>
```

The following table specifies additional information for the attributes of the **ShrinkDBTaskAttributeGroup** attribute group.

| Attribute | Description |
|-----------|-------------|
| DatabaseSizeLimit | An integer that specifies the size limit of the database in megabytes. |
| DatabasePercentLimit | An integer that specifies the limit of free space in units of percentage. When free space in the database falls below the percentage represented by the integer value, shrinking of the database stops. |
| DatabaseReturnFreeSpace | An enumeration value that specifies whether to retain free space in the database file or to return it to the operating system. "True" specifies that free space is returned to the operating system. "False" specifies that free space is retained in the database file. |

### 2.7.1.11.1.1.1.19    Execute TSQL Task

The Execute TSQL Task has no attributes that it adds to the SqlTaskData element. There MUST NOT be any attributes for the **SqlTaskData** element beyond those that are specified in the SqlTaskBaseAttributeGroup attribute group and the MaintenanceTaskBaseAttributeGroup attribute group when the **ExecutableType** attribute of the containing Executable element is equal to one of the following values:

▪ Microsoft.SqlServer.Management.DatabaseMaintenance.DbMaintenanceTSQLExecuteTask, Microsoft.SqlServer.MaintenancePlanTasks, Version=11.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91 (for DTSX2 2012/01)

▪ "Microsoft.DbMaintenanceTSQLExecuteTask (for DTSX2 2014/01)

### 2.7.1.11.1.1.1.20    UpdateStatisticsTaskAttributeGroup

The **UpdateStatisticsTaskAttributeGroup** attribute group specifies the attributes that are used by the SqlTaskData element when the **ExecutableType** attribute of the containing Executable element is equal to one of the following values:

▪ Microsoft.SqlServer.Management.DatabaseMaintenance.DbMaintenanceUpdateStatisticsTask, Microsoft.SqlServer.MaintenancePlanTasks, Version=11.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91 (for DTSX2 2012/01)

▪ Microsoft.DbMaintenanceUpdateStatisticsTask (for DTSX2 2014/01)

The attributes that are specified in this section MUST NOT be used unless the **ExecutableType** attribute is equal to this value.

The following is the XSD for the **UpdateStatisticsTaskAttributeGroup** attribute group.

```
<xs:attributeGroup name="UpdateStatisticsTaskAttributeGroup">
  <xs:attribute name="UpdateStatisticsType"
                type="SQLTask:SqlTaskUpdateStatisticsTypeEnum"/>
  <xs:attribute name="UpdateScanType" type="SQLTask:SqlTaskUpdateScanTypeEnum"/>
  <xs:attribute name="UpdateSampleValue" type="xs:int"/>
</xs:attributeGroup>
```

The following table specifies additional information for the attributes of the **UpdateStatisticsTaskAttributeGroup** attribute group.

| Attribute | Description |
|---|---|
| UpdateStatisticsType | An enumeration value that specifies the database objects that have updated statistics. |
| UpdateScanType | An enumeration value that specifies the type of scan to perform for statistics gathering. |
| UpdateSampleValue | An integer value for which the interpretation is based upon the value of the **UpdateScanType** attribute. This integer value can represent a percentage or a count of rows. |

#### 2.7.1.11.1.1.1.20.1 SqlTaskUpdateScanTypeEnum

The **SqlTaskUpdateScanTypeEnum** type contains the enumeration values for the **UpdateScanType** attribute.

The following is the XSD for the **SqlTaskUpdateScanTypeEnum** type.

```
<xs:simpleType name="SqlTaskUpdateScanTypeEnum">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="1"/>
    <xs:maxInclusive value="3"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **SqlTaskUpdateScanTypeEnum** type.

| Enumeration value | Description |
|---|---|
| 1 | Specifies that the update is performed on a sample by percentage. |
| 2 | Specifies that the update is performed on a sample by rows. |
| 3 | Specifies that the updates are performed on a full scan. |

#### 2.7.1.11.1.1.1.20.2 SqlTaskUpdateStatisticsTypeEnum

The **SqlTaskUpdateStatisticsTypeEnum** type contains the enumeration values for the **UpdateStatisticsType** attribute.

The following is the XSD for the **SqlTaskUpdateStatisticsTypeEnum** type.

```
<xs:simpleType name="SqlTaskUpdateStatisticsTypeEnum">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="2"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **SqlTaskUpdateStatisticsTypeEnum** type.

| Enumeration value | Description |
|---|---|
| 0 | Specifies that statistics are updated for indexes only. |

| Enumeration value | Description |
|---|---|
| 1 | Specifies that only column statistics are updated. |
| 2 | Specifies that all existing statistics are updated. |

### 2.7.1.12 FileSystemDataObjectDataType

The **FileSystemDataObjectDataType** complex type is used to specify settings and parameters that are specific to the File System Task.

The following is the XSD for the **FileSystemDataObjectDataType** complex type.

```
<xs:complexType name="FileSystemDataObjectDataType">
  <xs:sequence/>
  <xs:attribute name="TaskOperationType" form="unqualified"
                type="DTS:FileSystemTaskOperationTypeEnum"
                use="optional" default="CopyFile"/>
  <xs:attribute name="OperationName" form="unqualified"
                type="xs:string" use="optional" default=""/>
  <xs:attribute name="TaskOverwriteDestFile" form="unqualified"
                use="optional" default="False">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="False"/>
        <xs:enumeration value="True"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="TaskSourcePath" form="unqualified"
                type="xs:string" use="optional" default=""/>
  <xs:attribute name="TaskIsSourceVariable" form="unqualified"
                use="optional" default="False">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="False"/>
        <xs:enumeration value="True"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="TaskDestinationPath" form="unqualified"
                type="xs:string" use="optional" default=""/>
  <xs:attribute name="TaskIsDestinationVariable" form="unqualified">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="False"/>
        <xs:enumeration value="True"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="TaskFileAttributes" form="unqualified"
                type="DTS:FileSystemTaskFileAttributesEnum"
                use="optional" default="Normal"/>
</xs:complexType>
```

The **FileSystemDataObjectDataType** complex type MUST NOT have child elements.

The following table provides additional information about the attributes in the **FileSystemDataObjectDataType** complex type.

| Attribute | Description |
|---|---|
| TaskOperationType | This enumeration value specifies the type of file system task that is performed. |
| OperationName | This string value contains the name given to the operation. |
| TaskOverwriteDestFile | This enumeration value specifies whether the destination file can be overwritten. <br> "True" specifies that the destination file can be overwritten. <br> "False" specifies that the destination file cannot be overwritten. |
| TaskSourcePath | This string value represents the source file path. It contains either the **DTSID** property of a defined **ConnectionManager** object or a string that contains the name of a variable. |
| TaskIsSourceVariable | This enumeration value specifies whether a variable or a file connection manager specifies the source path. <br> "True" specifies that the **TaskSourcePath** is a variable. <br> "False" specifies that the **TaskSourcePath** is a file connection manager. |
| TaskDestinationPath | This string value represents the destination file path. It contains either the **DTSID** property of a defined **ConnectionManager** object or a string that contains the name of a variable. |
| TaskIsDestinationVariable | This enumeration value specifies whether a variable or a file connection manager specifies the destination path. <br> "True" specifies that the **TaskDestinationPath** value is a variable. <br> "False" specifies that the **TaskDestinationPath** value is a file connection manager. |
| TaskFileAttributes | This enumeration value specifies the file attributes for the file system file. |

### 2.7.1.12.1 FileSystemTaskOperationTypeEnum

The **FileSystemTaskOperationTypeEnum** type contains the allowed enumeration values for the **FileSystemTaskOperationType** attribute of the FileSystemDataObjectDataType type.

The following is the XSD for the **FileSystemTaskOperationTypeEnum** type.

```
<xs:simpleType name="FileSystemTaskOperationTypeEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="CopyFile" />
    <xs:enumeration value="CreateDirectory" />
    <xs:enumeration value="DeleteDirectory" />
    <xs:enumeration value="DeleteFile" />
    <xs:enumeration value="MoveDirectory" />
    <xs:enumeration value="MoveFile" />
    <xs:enumeration value="RenameFile" />
    <xs:enumeration value="SetAttributes" />
    <xs:enumeration value="DeleteDirectoryContent" />
    <xs:enumeration value="CopyDirectory" />
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **FileSystemTaskOperationTypeEnum** type.

| Enumeration value | Description |
|---|---|
| CopyFile | Specifies that the task operation copies a file. |

| Enumeration value | Description |
|---|---|
| CreateDirectory | Specifies that the task operation creates a directory. |
| DeleteDirectory | Specifies that the task operation deletes a directory. |
| DeleteFile | Specifies that the task operation deletes a file. |
| MoveDirectory | Specifies that the task operation moves a directory. |
| MoveFile | Specifies that the task operation moves a file. |
| RenameFile | Specifies that the task operation renames a file. |
| SetAttributes | Specifies that the task operation sets attributes. |
| DeleteDirectoryContent | Specifies that the task operation deletes directory contents. |
| CopyDirectory | Specifies that the task operation copies a directory. |

### 2.7.1.12.2    FileSystemTaskFileAttributesEnum

The **FileSystemTaskFileAttributesEnum** type contains the allowed enumeration values for the **TaskFileAttributes** attribute of the FileSystemDataObjectDataType type.

The following is the XSD for the **FileSystemTaskFileAttributesEnum** type.

```
<xs:simpleType name="FileSystemTaskFileAttributesEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Archive" />
    <xs:enumeration value="Hidden" />
    <xs:enumeration value="Normal" />
    <xs:enumeration value="ReadOnly" />
    <xs:enumeration value="System" />
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **FileSystemTaskFileAttributesEnum** type.

| Enumeration value | Description |
|---|---|
| Archive | This enumeration value specifies that the **Archive** attribute of the file MUST be set by the task. |
| Hidden | This enumeration value specifies that the **Hidden** attribute of the file MUST be set by the task. |
| Normal | This enumeration value specifies that the file is normal and that none of the file attributes MUST be set by the task. |
| ReadOnly | This enumeration value specifies that the **ReadOnly** attribute of the file MUST be set by the task. |
| System | This enumeration value specifies that the **System** attribute of the file MUST be set by the task. |

### 2.7.1.13 FtpDataObjectDataType

The **FtpDataObjectDataType** complex type is used to specify parameters, properties, and settings that are specific to the FTP Task.

The following is the XSD of the **FtpDataObjectDataType** complex type.

```xml
<xs:complexType name="FtpDataObjectDataType">
  <xs:sequence>
  </xs:sequence>
  <xs:attribute name="ConnectionName" type="xs:string"
              use="optional" default="" form="unqualified"/>
  <xs:attribute name="StopOnFailure" form="unqualified">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="False"/>
        <xs:enumeration value="True"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="TaskOperationType"
              type="DTS:FtpDataTaskOperationTypeEnum"
              use="optional" default="Send" form="unqualified"/>
  <xs:attribute name="TaskOperationName" type="xs:string"
              use="optional" default="" form="unqualified"/>
  <xs:attribute name="TaskIsLocalPathVar" form="unqualified"
              use="optional" default="False">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="False"/>
        <xs:enumeration value="True"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="TaskLocalPath" type="xs:string"
              use="optional" default="" form="unqualified"/>
  <xs:attribute name="TaskIsRemotePathVar" form="unqualified"
              use="optional" default="False">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="False"/>
        <xs:enumeration value="True"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="TaskRemotePath" type="xs:string"
              use="optional" default="" form="unqualified"/>
  <xs:attribute name="TaskOverwriteDest" form="unqualified"
              use="optional" default="False">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="False"/>
        <xs:enumeration value="True"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="TaskTransferAscii" form="unqualified"
              use="optional" default="False">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="False"/>
        <xs:enumeration value="True"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:complexType>
```

The **FtpDataObjectDataType** complex type MUST NOT contain elements.

The following table specifies the attributes of the **FtpDataObjectDataType** complex type.

| Attribute | Description |
|---|---|
| ConnectionName | A string value that specifies a reference to a connection manager by its **DTSID** property value or by its **ObjectName** property value. |
| StopOnFailure | An enumeration value that specifies whether the task fails if the FTP operation fails. "True" specifies that the task fails if the FTP operation fails. "False" specifies that the task does not fail if the FTP operation fails. |
| TaskOperationType | An enumeration value that specifies the FTP operation that the task performs. |
| TaskOperationName | A string value that specifies a name that is given to the operation. |
| TaskIsLocalPathVar | An enumeration value that specifies whether the **TaskLocalPath** attribute contains the name of a variable. "True" specifies that the **TaskLocalPath** attribute contains a variable name. "False" specifies that the **TaskLocalPath** attribute does not contain a variable name. |
| TaskLocalPath | A string value that specifies the path in the local location, or if the **TaskIsLocalPathVar** attribute is equal to "True", a string value that specifies the name of the variable. |
| TaskIsRemotePathVar | An enumeration value that specifies whether the **TaskRemotePath** attribute contains the name of a variable. "True" specifies that the **TaskRemotePath** attribute contains a variable name. "False" specifies that the **TaskRemotePath** attribute does not contain a variable name. |
| TaskRemotePath | A string value that specifies the path in the remote location, or if the **TaskIsRemotePathVar** attribute is equal to "True", a string value that specifies the name of the variable. |
| TaskOverwriteDest | An enumeration value specifies whether the FTP operation can overwrite objects at the destination. "True" specifies that the task can overwrite objects at the destination. "False" specifies that the task cannot overwrite objects at the destination. |
| TaskTransferAscii | An enumeration value specifies whether file transfer is performed in ASCII mode. "True" specifies that file transfer is performed in ASCII mode. "False" specifies that file transfer is not performed in ASCII mode. |

### 2.7.1.13.1    FtpDataTaskOperationTypeEnum

The **FtpDataTaskOperationTypeEnum** type contains the enumeration values for specifying what FTP operation is performed.

The following is the XSD for the **FtpDataTaskOperationTypeEnum** type.

```
<xs:simpleType name="FtpDataTaskOperationTypeEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="DeleteLocal"/>
    <xs:enumeration value="DeleteRemote"/>
    <xs:enumeration value="MakeDirLocal"/>
    <xs:enumeration value="MakeDirRemote"/>
    <xs:enumeration value="Receive"/>
    <xs:enumeration value="RemoveDirLocal"/>
```

```
            <xs:enumeration value="RemoveDirRemote"/>
            <xs:enumeration value="Send"/>
          </xs:restriction>
      </xs:simpleType>
```

The following table specifies the enumeration values for the **FtpDataTaskOperationTypeEnum** type.

| Enumeration value | Description |
| --- | --- |
| DeleteLocal | Specifies that the FTP operation deletes a local file. |
| DeleteRemote | Specifies that the FTP operation deletes a remote file. |
| MakeDirLocal | Specifies that the FTP operation creates a local directory. |
| MakeDirRemote | Specifies that the FTP operation creates a remote directory. |
| Receive | Specifies that the FTP operation receives a file or directory. |
| RemoveDirLocal | Specifies that the FTP operation removes a local directory. |
| RemoveDirRemote | Specifies that the FTP operation removes a remote directory. |
| Send | Specifies that the FTP operation sends a file or directory. |

### 2.7.1.14    MessageQueueTaskData Element

The **MessageQueueTaskData** element is used to specify parameters, properties, and settings that are specific to the execution of a Message Queue Task. The **MessageQueueTaskData** element resides in the MessageQueueTask namespace. The ExecutableObjectDataType type refers to the **MessageQueueTaskData** element within an XSD **choice** element, using the XSD **ref** attribute, in the following XSD fragment.

```
            <xs:element ref="MessageQueueTask:MessageQueueTaskData"/>
```

### 2.7.1.14.1    MessageQueueTask Namespace

The **MessageQueueTask** namespace specifies the elements and types that are used by the Message Queue Task.

The following is the XSD declaration for the **MessageQueueTask** namespace.

```
    xmlns:MessageQueueTask="www.microsoft.com/sqlserver/dts/tasks/messagequeuetask"
```

### 2.7.1.14.1.1  MessageQueueTaskData Element

The following is the XSD for the element declaration for the **MessageQueueTaskData** element in the MessageQueueTask namespace.

```
    <xs:element name="MessageQueueTaskData" type="MessageQueueTask:MessageQueueTaskType"/>
```

### 2.7.1.14.1.1.1        MessageQueueTaskType

The **MessageQueueTaskType** complex type holds parameters, properties, and settings for the Message Queue Task.

The following is the XSD for the **MessageQueueTaskType** complex type.

```
<xs:complexType name="MessageQueueTaskType">
  <xs:sequence>
  </xs:sequence>
  <xs:attribute name="CurrentPackageID" type="DTS:uuid" use="optional" default=""/>
  <xs:attribute name="TaskDescription" type="xs:string" use="optional" default=""/>
  <xs:attribute name="DataFileOverWritable" use="optional" default="False">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="False"/>
        <xs:enumeration value="True"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="DTSMessageLineageID" type="xs:string" use="optional" default=""/>
  <xs:attribute name="DTSMessagePackageID" type="xs:string" use="optional" default=""/>
  <xs:attribute name="DTSMessageVersionID" type="xs:string" use="optional" default=""/>
  <xs:attribute name="ErrorIfMessageTimeOut" use="optional" default="False">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="False"/>
        <xs:enumeration value="True"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="QueuePath" type="xs:string"/>
  <xs:attribute name="ReceiveMessageTimeOut" type="xs:int" use="optional"
              default="0"/>
  <xs:attribute name="ReceiveMessageType"
              type="MessageQueueTask:MSMQMessageTypeEnum"
              use="optional" default="DTSMQMessageType_String"/>
  <xs:attribute name="RemoveFromQueue" use="optional" default="False">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="False"/>
        <xs:enumeration value="True"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="SaveDataFileName" type="xs:string" use="optional"
              default=""/>
  <xs:attribute name="StringCompareType"
              type="MessageQueueTask:MSMQStringCompareTypeEnum"
              use="optional" default="DTSMQStringMessageCompare_None"/>
  <xs:attribute name="StringCompareValue" type="xs:string" use="optional"
              default=""/>
  <xs:attribute name="StringMessageToVariableName" type="xs:string"
              use="optional" default=""/>
  <xs:attribute name="ReceiveVariableMessage" type="xs:string"
              use="optional" default=""/>
  <xs:attribute name="TaskType"
              type="MessageQueueTask:MSMQTaskTypeEnum"
              use="optional" default="DTSMQType_Sender"/>
  <xs:attribute name="UseEncryption" use="optional" default="False">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="False"/>
        <xs:enumeration value="True"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="EncryptionAlgorithm"
              type="MessageQueueTask:MSMQEncryptionAlgorithmEnum"
              use="optional" default="Rc2"/>
```

```
        <xs:attribute name="Use2000Format" use="optional" default="False">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="False"/>
              <xs:enumeration value="True"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="DataFileName" type="xs:string" use="optional" default=""/>
        <xs:attribute name="StringMessage" type="xs:string" use="optional" default=""/>
        <xs:attribute name="VariableMessage" type="xs:string" use="optional" default=""/>
        <xs:attribute name="MessageType"
                      type="MessageQueueTask:MSMQMessageTypeEnum"
                      use="optional" default="DTSMQMessageType_String"/>
    </xs:complexType>
```

The **MessageQueueTaskType** complex type MUST NOT contain elements.

The following table specifies additional information for the attributes of the **MessageQueueTaskType** complex type.

| Attribute | Description |
|---|---|
| CurrentPackageID | A UUID value that specifies the **DTSID** attribute of the currently executing package. |
| TaskDescription | A string value that specifies a description for the task. |
| DataFileOverWritable | An enumeration value that specifies whether the message can overwrite an existing file. This enumeration value applies only if the **MessageType** attribute is equal to "DTSMQMessageType_DataFile". <br> "True" specifies that the task can overwrite a file. <br> "False" specifies that the task cannot overwrite a file. |
| DTSMessageLineageID | A string value.<98> |
| DTSMessagePackageID | A UUID value that specifies the **DTSID** property of the package that is the source of the message. |
| DTSMessageVersionID | A string value that specifies the version of the message. |
| ErrorIfMessageTimeOut | An enumeration value that specifies whether the task fails when the message times out. <br> "True" specifies that the task MUST fail when the message times out. <br> "False" specifies that the task does not necessarily fail when the message times out. |
| QueuePath | A string value that specifies the **ObjectName** property of a connection manager for this message queue. |
| ReceiveMessageTimeOut | An integer value that specifies the number of seconds before a received message times out. |
| ReceiveMessageType | An enumeration value that specifies the type of the received message. |
| RemoveFromQueue | An enumeration value that specifies whether to remove the message from the queue. <br> "True" specifies that the message MUST be removed from the queue. <br> "False" specifies that the message MUST NOT be removed from the queue. |
| SaveDataFileName | A string value that specifies the name of the file to which an incoming message is saved. |

| Attribute | Description |
|---|---|
| StringCompareType | An enumeration value that specifies the string comparison method. This enumeration value is used only when the **TaskType** attribute is set to value "DTSMQType_Receiver". |
| StringCompareValue | A string value that specifies a string to which a string message is compared when the **TaskType** attribute is set to "DTSMQType_Receiver". |
| StringMessageToVariableName | A string value that specifies the name of the variable to which the message contents are saved. |
| ReceiveVariableMessage | A string value that specifies the name of the variable where a received message is saved. |
| TaskType | An enumeration value that specifies whether the task is sending a message or receiving a message. |
| UseEncryption | An enumeration value that specifies whether to encrypt the messages.<br>"True" specifies that the message MUST be encrypted.<br>"False" specifies that the message MUST NOT be encrypted. |
| EncryptionAlgorithm | An enumeration value that specifies the encryption algorithm to use. |
| Use2000Format | An enumeration value that specifies whether the message is formatted in a specific format.<99> |
| DataFileName | A string value that specifies the name of the file that is used if the message type uses a file. |
| StringMessage | A string value that contains the message text. This string value is used only if the **MessageType** attribute is set to the "DTSMQMessageType_String" value. |
| VariableMessage | A string value that contains the message text. This string value is used only if the **MessageType** attribute is set to the "DTSMQMessageType_Variables" value. |
| MessageType | An enumeration value that specifies the type of message that is sent. This enumeration value is used only when the **TaskType** attribute is set to "DTSMQType_Sender". |

#### 2.7.1.14.1.1.1.1 MSMQStringCompareTypeEnum

The **MSMQStringCompareTypeEnum** type contains the enumeration values for the **StringCompareType** attribute.

The following is the XSD of the **MSMQStringCompareTypeEnum** type.

```
<xs:simpleType name="MSMQStringCompareTypeEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="DTSMQStringMessageCompare_Contains"/>
    <xs:enumeration value="DTSMQStringMessageCompare_Exact"/>
    <xs:enumeration value="DTSMQStringMessageCompare_IgnoreCase"/>
    <xs:enumeration value="DTSMQStringMessageCompare_None"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **MSMQStringCompareTypeEnum** type.

| Enumeration value | Description |
|---|---|
| DTSMQStringMessageCompare_Contains | Specifies that the comparison operation is successful if the received message contains the comparison string. |
| DTSMQStringMessageCompare_Exact | Specifies that the comparison operation is successful if the received message exactly matches the comparison string. |
| DTSMQStringMessageCompare_IgnoreCase | Specifies that the comparison operation is successful if the received message matches the comparison string, ignoring the case of letters. |
| DTSMQStringMessageCompare_None | Specifies that no comparison operation is performed. |

### 2.7.1.14.1.1.1.2    MSMQTaskTypeEnum

The **MSMQTaskTypeEnum** type contains the enumeration values for the **TaskType** attribute.

The following is the XSD of the **MSMQTaskTypeEnum** type.

```
<xs:simpleType name="MSMQTaskTypeEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="DTSMQType_Receiver"/>
    <xs:enumeration value="DTSMQType_Sender"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **MSMQTaskTypeEnum** type.

| Enumeration value | Description |
|---|---|
| DTSMQType_Receiver | Specifies that the task receives messages. |
| DTSMQType_Sender | Specifies that the task sends messages. |

### 2.7.1.14.1.1.1.3    MSMQMessageTypeEnum

The **MSMQMessageTypeEnum** type contains the enumeration values for the **MessageType** attribute.

The following is the XSD of the **MSMQMessageTypeEnum** type.

```
<xs:simpleType name="MSMQMessageTypeEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="DTSMQMessagType_StringMessageToVariable"/>
    <xs:enumeration value="DTSMQMessageType_DataFile"/>
    <xs:enumeration value="DTSMQMessageType_String"/>
    <xs:enumeration value="DTSMQMessageType_Variables"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **MSMQMessageTypeEnum** type.

| Enumeration value | Description |
|---|---|
| DTSMQMessageType_StringMessageToVariable | Specifies that the message source is text and that the message destination is a variable. This value is valid only if the **TaskType** attribute is equal to "DTSMQType_Sender". |
| DTSMQMessageType_DataFile | Specifies that the message source is a data file. |
| DTSMQMessageType_String | Specifies that the message source is a string. |
| DTSMQMessageType_Variables | Specifies that the message source is a variable and that the message destination is also a variable. |

#### 2.7.1.14.1.1.1.4    MSMQEncryptionAlgorithmEnum

The **MSMQEncryptionAlgorithmEnum** type contains the enumeration values for the EncryptionAlgorithm attribute.

The following is the XSD of the **MSMQEncryptionAlgorithmEnum** type.

```
<xs:simpleType name="MSMQEncryptionAlgorithmEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Rc2"/>
    <xs:enumeration value="Rc4"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **MSMQEncryptionAlgorithmEnum** type.

| Enumeration value | Description |
|---|---|
| Rc2 | Specifies that the RC2 encryption algorithm [RFC2268] is used. |
| Rc4 | Specifies that the RC4 encryption algorithm [SCHNEIER] is used. |

### 2.7.1.15    ScriptProjectObjectDataType

The **ScriptProjectObjectDataType** complex type is used to specify parameters, properties, and settings that are specific to the Script Task.

The following is the XSD of the **ScriptProjectObjectDataType** complex type.

```
<xs:complexType name="ScriptProjectObjectDataType">
  <xs:sequence>
    <xs:element name="ProjectItem"
                type="DTS:ScriptProjectProjectItemType"
                minOccurs="0" maxOccurs="unbounded"
                form="unqualified"/>
    <xs:element name="BinaryItem"
                type="DTS:ScriptProjectBinaryItemType"
                minOccurs="0" form="unqualified"/>
    <xs:element name="Breakpoint"
                type="DTS:ScriptProjectBreakpointType"
                minOccurs="0" maxOccurs="unbounded"
                form="unqualified"/>
  </xs:sequence>
```

```
            <xs:attribute name="Name" type="xs:string" form="unqualified"/>
            <xs:attribute name="VSTAMajorVersion" type="xs:int" form="unqualified"/>
            <xs:attribute name="VSTAMinorVersion" type="xs:int" form="unqualified"/>
            <xs:attribute name="Language" type="DTS:ScriptProjectLanguageEnum"
                          form="unqualified"/>
            <xs:attribute name="EntryPoint" type="xs:string" form="unqualified"/>
            <xs:attribute name="ReadOnlyVariables" type="xs:string" form="unqualified"/>
            <xs:attribute name="ReadWriteVariables" type="xs:string" form="unqualified"/>
        </xs:complexType>
```

The following table provides additional information about the elements, types, and constraints in the **ScriptProjectObjectDataType** complex type.

| Element | Additional constraints | Type definition | Description |
|---------|------------------------|-----------------|-------------|
| ProjectItem | None | ScriptProjectProjectItemType | A string value that specifies the source code for a single item of a development environment project. |
| BinaryItem | None | ScriptProjectBinaryItemType | A base64 binary value that specifies the compiled binary from the development environment of the script project. |
| Breakpoint | None | ScriptProjectBreakpointType | A complex value that specifies a breakpoint in the code. |

The following table specifies the attributes of the **ScriptProjectObjectDataType** complex type.

| Attribute | Description |
|-----------|-------------|
| Name | A string value that specifies the name of the script project task. |
| VSTAMajorVersion | An integer value that specifies the major version number.<100> |
| VSTAMinorVersion | An integer value that specifies the minor version number.<101> |
| Language | An enumeration value that specifies the language for the script project. |
| EntryPoint | A string value that specifies the name of the entry point routine in the code for the script project. |
| ReadOnlyVariables | A string value that specifies read-only variables in the package file that are available to the code in the script project. |
| ReadWriteVariables | A string value that specifies the read-write variables in the package file that are available to the code in the script project. |

### 2.7.1.15.1    ScriptProjectProjectItemType

The **ScriptProjectProjectItemType** complex type is used to specify the source code for the Script Component that is executed.

The following is the XSD of the **ScriptProjectProjectItemType** complex type.

```
        <xs:complexType name="ScriptProjectProjectItemType">
          <xs:simpleContent>
            <xs:extension base="xs:string">
              <xs:attribute name="Name" type="xs:string" form="unqualified"/>
              <xs:attribute name="Encoding" type="xs:string" form="unqualified"/>
```

```
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
```

The value of element of the **ScriptProjectProjectItemType** is a string that specifies one source code item of the source code for the Script Component that is executed.

The following table specifies additional information for the attributes of the **ScriptProjectProjectItemType** complex type.

| Attribute | Description |
|---|---|
| Name | A string value that specifies the name of the source file in the development environment in which the source code was developed. |
| Encoding | A string value that specifies the name of the encoding that is used to encode the source code for the Script Component that is executed. |

### 2.7.1.15.2    ScriptProjectBinaryItemType

The **ScriptProjectBinaryItemType** complex type contains a representation of the binary of the compiled code for the Script Component.

The following is the XSD of the **ScriptProjectBinaryItemType** complex type.

```
<xs:complexType name="ScriptProjectBinaryItemType">
  <xs:simpleContent>
    <xs:extension base="xs:base64Binary">
      <xs:attribute name="Name" type="xs:string" form="unqualified"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

The value of element of the **ScriptProjectBinaryItemType** complex type is a base64 binary representation that specifies the executable code for the Script Component.

The following table specifies additional information for the attributes of the **ScriptProjectBinaryItemType** complex type.

| Attribute | Description |
|---|---|
| Name | A string value that specifies the name of the binary file in the development environment in which the source code was developed. |

### 2.7.1.15.3    ScriptProjectBreakpointType

The **ScriptProjectBreakpointType** complex type contains a representation of the breakpoints that are set in the development environment for the Script Component.

The following is the XSD of the **ScriptProjectBreakpointType** complex type.

```
<xs:complexType name="ScriptProjectBreakpointType">
  <xs:sequence/>
  <xs:attribute name="Name" type="xs:string" form="unqualified"/>
```

```
            <xs:attribute name="Enabled" type="xs:string" form="unqualified"/>
            <xs:attribute name="Function" type="xs:string" form="unqualified"/>
            <xs:attribute name="File" type="xs:string" form="unqualified"/>
            <xs:attribute name="Line" type="xs:int" form="unqualified"/>
            <xs:attribute name="Column" type="xs:int" form="unqualified"/>
            <xs:attribute name="Condition" type="xs:string" form="unqualified"/>
            <xs:attribute name="ConditionType" type="DTS:ScriptProjectConditionTypeEnum"
                    form="unqualified"/>
            <xs:attribute name="Language" type="xs:string" form="unqualified"/>
            <xs:attribute name="HitCount" type="xs:int" form="unqualified"/>
            <xs:attribute name="HitCountType" type="DTS:ScriptProjectHitCountTypeEnum"
                    form="unqualified"/>
    </xs:complexType>
```

The following table specifies additional information for the attributes of the
**ScriptProjectBinaryItemType** complex type.

| Attribute | Description |
|---|---|
| Name | A string value that specifies the name of the breakpoint. |
| Enabled | An enumeration value that specifies whether the breakpoint is enabled. "True" specifies that the breakpoint is enabled. "False" specifies that the breakpoint is not enabled. |
| Function | A string value that specifies the name of the function in which the breakpoint is set. |
| File | A string value that specifies the name of the file in which the breakpoint is set. |
| Line | An integer value that specifies the line number at which the breakpoint is set. |
| Column | An integer value that specifies the character column at which the breakpoint is set. |
| Condition | A string value that specifies a condition that can be evaluated as a Boolean to determine whether the breakpoint is fired. Whether the breakpoint is fired depends on this **Condition** value and on the values of the following attributes: **ConditionType** **HitCount** **HitCountType** |
| ConditionType | An enumeration value that specifies when the evaluation of the string in the **Condition** attribute causes a breakpoint to be fired. |
| Language | A string value that specifies the language of the Script Component.<102> |
| HitCount | An integer value that specifies a criteria for a hit count. This value, along with the value of the **HitCountType** attribute, is used to determine whether a break point is fired. |
| HitCountType | An enumeration value that specifies the type of hit count condition that causes the breakpoint to be fired. |

### 2.7.1.15.4    ScriptProjectLanguageEnum

The **ScriptProjectLanguageEnum** type contains the enumeration values for the Language attribute.

The following is the XSD of the **ScriptProjectLanguageEnum** type.

```
    <xs:simpleType name="ScriptProjectLanguageEnum">
      <xs:restriction base="xs:string">
```

```
      <xs:enumeration value="CSharp"/>
      <xs:enumeration value="VisualBasic"/>
    </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **ScriptProjectLanguageEnum** type.

| Enumeration value | Description |
|---|---|
| CSharp | Specifies that the C# language [MSDN-C#LS] is used. |
| VisualBasic | Specifies that the Visual Basic language [MSDN-VBLR] is used. |

### 2.7.1.15.5    ScriptProjectHitCountTypeEnum

The **ScriptProjectHitCountTypeEnum** type contains the enumeration values for the **HitCountType** attribute.

The following is the XSD of the **ScriptProjectHitCountTypeEnum** type.

```
<xs:simpleType name="ScriptProjectHitCountTypeEnum">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="1"/>
    <xs:maxInclusive value="4"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **ScriptProjectHitCountTypeEnum** type.

| Enumeration value | Description |
|---|---|
| 1 | Specifies that there is no hit count. |
| 2 | Specifies that the hit count is equal to the specified condition. |
| 3 | Specifies that the hit count is greater than or equal to the specified condition. |
| 4 | Specifies that there are multiple hit count conditions. |

### 2.7.1.15.6    ScriptProjectConditionTypeEnum

The **ScriptProjectConditionTypeEnum** type contains the enumeration values for the **ConditionType** attribute.

The following is the XSD of the **ScriptProjectConditionTypeEnum** type.

```
<xs:simpleType name="ScriptProjectConditionTypeEnum">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="1"/>
    <xs:maxInclusive value="2"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **ScriptProjectConditionTypeEnum** type.

| Enumeration value | Description |
|---|---|
| 1 | Break when condition is true. |
| 2 | Break when condition is changed. |

## 2.7.1.16    SendMailTaskData Element

The **SendMailTaskData** element is used to specify parameters, properties, and settings that are specific to the execution of a Send Mail Task. This element resides in the SendMailTask namespace. The ExecutableObjectDataType type refers to this element within an XSD **choice** element, using the XSD **ref** attribute, in the following XSD fragment.

```
<xs:element ref="SendMailTask:SendMailTaskData"/>
```

### 2.7.1.16.1    SendMailTask Namespace

The **SendMailTask** namespace specifies elements and types that are used by the Send Mail Task.

The following is the XSD declaration for the **SendMailTask** namespace.

```
xmlns:SendMailTask="www.microsoft.com/sqlserver/dts/tasks/sendmailtask"
```

#### 2.7.1.16.1.1  SendMailTaskData Element

The following is the XSD for the element declaration for the **SendMailTaskData** element in the SendMailTask namespace.

```
<xs:element name="SendMailTaskData" type="SendMailTask:SendMailTaskType"/>
```

##### 2.7.1.16.1.1.1    SendMailTaskType

The **SendMailTaskType** complex type holds parameters, properties, and settings for the Send Mail Task.

The following is the XSD for the **SendMailTaskType** complex type.

```
<xs:complexType name="SendMailTaskType">
  <xs:sequence>
    <xs:element name="Attachments" minOccurs="0"
                type="SendMailTask:AttachmentsElementType"/>
  </xs:sequence>
  <xs:attribute name="SMTPServer" type="xs:string" use="optional" default=""/>
  <xs:attribute name="From" type="xs:string" use="optional" default=""/>
  <xs:attribute name="To" type="xs:string" use="optional" default=""/>
  <xs:attribute name="CC" type="xs:string" use="optional" default=""/>
  <xs:attribute name="BCC" type="xs:string" use="optional" default=""/>
  <xs:attribute name="Subject" type="xs:string" use="optional" default=""/>
  <xs:attribute name="Priority" type="SendMailTask:SendMailPriorityEnum"
                use="optional" default="Normal"/>
  <xs:attribute name="MessageSourceType" type="DTS:SourceTypeEnum"/>
  <xs:attribute name="MessageSource" type="xs:string" use="optional"
                default="DirectInput"/>
</xs:complexType>
```

The following table provides additional information about the elements, types, and constraints for the **SendMailTaskType** complex type.

| Element | Additional constraints | Type definition | Description |
|---|---|---|---|
| Attachments | None. | AttachmentsElementType | Contains information about the files to be attached to the message. |

The following table specifies additional information for the attributes of the **SendMailTaskType** complex type.

| Attribute | Description |
|---|---|
| SMTPServer | A string value that specifies a reference to a connection manager by its **DTSID** property value or by its **ObjectName** property value. |
| From | A string value that specifies the From field of the mail message. |
| To | A string value that specifies the To field of the mail message. |
| CC | A string value that specifies the Cc field of the mail message. |
| BCC | A string value that specifies the Bcc field of the mail message. |
| Subject | A string value that specifies the Subject field of the mail message. |
| Priority | An enumeration value that specifies the priority of the mail message. |
| MessageSourceType | An enumeration value that specifies the source type of the mail message. |
| MessageSource | A string value that specifies the content of the mail message. This value is used only if the **MessageSourceType** attribute is set to the value "DirectInput". |

#### 2.7.1.16.1.1.1.1    AttachmentsElementType

The **AttachmentsElementType** complex type is used to specify attachments to mail messages that are sent with the Send Mail Task.

The following is the XSD for the **AttachmentsElementType** complex type.

```
<xs:complexType name="AttachmentsElementType">
  <xs:sequence/>
  <xs:attribute name="FileName" type="xs:string"/>
</xs:complexType>
```

The following table specifies additional information for the attributes of the **AttachmentsElementType** complex type.

| Attribute | Description |
|---|---|
| FileName | A string value that specifies the names of the files that are attached to the mail message that is sent by the Send Mail Task. If more than one file is attached, the file names MUST be separated by a vertical bar (\|) character. |

### 2.7.1.16.1.1.1.2    SendMailPriorityEnum

The **SendMailPriorityEnum** type contains the enumeration values for the **Priority** attribute.

The following is the XSD of the **SendMailPriorityEnum** type.

```
<xs:simpleType name="SendMailPriorityEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="High"/>
    <xs:enumeration value="Low"/>
    <xs:enumeration value="Normal"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **SendMailPriorityEnum** type.

| Enumeration value | Description |
|---|---|
| High | Specifies that the mail message is marked as high priority. |
| Low | Specifies that the mail message is marked as low priority. |
| Normal | Specifies that the mail message is marked as normal priority. |

### 2.7.1.17      TransferDatabaseTaskDataObjectDataType

The **TransferDatabaseTaskDataObjectDataType** complex type is used to specify parameters, properties, and settings that are specific to the Transfer Database Task.

The following is the XSD for the **TransferDatabaseTaskDataObjectDataType** complex type.

```
<xs:complexType name="TransferDatabasesTaskDataObjectDataType">
  <xs:sequence/>
  <xs:attribute name="SrcConn" type="xs:string" use="optional" default=""
                form="unqualified"/>
  <xs:attribute name="DestConn" type="xs:string" use="optional"
                default="" form="unqualified"/>
  <xs:attribute name="Action" type="DTS:TransferDatabaseTaskActionEnum"
                use="optional" default="0" form="unqualified"/>
  <xs:attribute name="Method" type="DTS:TransferDatabaseTaskMethodEnum"
                use="optional" default="0" form="unqualified"/>
  <xs:attribute name="SrcDBName" type="xs:string" use="optional"
                default="" form="unqualified"/>
  <xs:attribute name="ReattachSrcDB" type="xs:boolean"
                use="optional" default="false" form="unqualified"/>
  <xs:attribute name="SrcDBFiles" type="xs:string"
                use="optional" default="" form="unqualified"/>
  <xs:attribute name="DestDBName" type="xs:string"
                use="optional" default="" form="unqualified"/>
  <xs:attribute name="DestDBFiles" type="xs:string"
                use="optional" default="" form="unqualified"/>
  <xs:attribute name="DestOverwrite" type="xs:boolean"
                use="optional" default="false" form="unqualified"/>
</xs:complexType>
```

The **TransferDatabaseTaskDataObjectDataType** complex type MUST NOT contain any elements or values.

The following table specifies the attributes of the **TransferDatabaseTaskDataObjectDataType** complex type.

| Attribute | Description |
|---|---|
| SrcConn | A string value that specifies a reference to a connection manager by its **DTSID** property value or by its **ObjectName** property value. |
| DestConn | A string value that specifies a reference to a connection manager by its **DTSID** property value or by its **ObjectName** property value. |
| Action | An enumeration value that specifies which transfer action to perform. |
| Method | An enumeration value that specifies the method to use when performing the database transfer. |
| SrcDBName | A string value that specifies the name of the source database. |
| ReattachSrcDB | A Boolean value that specifies whether to attempt to reattach the source database if a failure occurs.<br>TRUE specifies that there MUST be an attempt to reattach the source database if a failure occurs.<br>FALSE specifies that there MUST NOT be an attempt to reattach the source database if a failure occurs. |
| SrcDBFiles | A string value that specifies the name and location of the files at the source database. The string value is a comma-delimited list. |
| DestDBName | A string value that specifies the name of the destination database. |
| DestDBFiles | A string value that specifies the name and location of the files at the destination database. The string value is a comma-delimited list. |
| DestOverwrite | A Boolean value that specifies whether to overwrite a database at the destination if a database of the same name exists at the destination.<br>TRUE specifies that a database of the same name at the destination MUST be overwritten.<br>FALSE specifies that a database of the same name at the destination MUST NOT be overwritten. |

### 2.7.1.17.1    TransferDatabaseTaskActionEnum

The **TransferDatabaseTaskActionEnum** type contains the enumeration values for the **Action** attribute.

The following is the XSD of the **TransferDatabaseTaskActionEnum** type.

```
<xs:simpleType name="TransferDatabaseTaskActionEnum">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="1"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **TransferDatabaseTaskActionEnum** type.

| Enumeration value | Description |
|---|---|
| 0 | The database MUST be transferred by using a copy operation. |

| Enumeration value | Description |
|---|---|
| 1 | The database MUST be transferred by using a move operation. |

### 2.7.1.17.2   TransferDatabaseTaskMethodEnum

The **TransferDatabaseTaskMethodEnum** type contains the enumeration values for the **Method** attribute.

The following is the XSD of the **TransferDatabaseTaskMethodEnum** type.

```
<xs:simpleType name="TransferDatabaseTaskMethodEnum">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="1"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **TransferDatabaseTaskMethodEnum** type.

| Enumeration value | Description |
|---|---|
| 0 | The database MUST be transferred when the database is offline. |
| 1 | The database MUST be transferred when the database is online. |

### 2.7.1.18       TransferErrorMessagesTaskDataObjectDataType

The **TransferErrorMessagesTaskDataObjectDataType** complex type is used to specify parameters, properties, and settings that are specific to the Transfer Error Messages Task.

The following is the XSD for the **TransferErrorMessagesTaskDataObjectDataType** complex type.

```
<xs:complexType name="TransferErrorMessagesTaskDataObjectDataType">
  <xs:sequence/>
  <xs:attribute name="TransferAllErrorMessags" type="xs:boolean"
                use="optional" default="false" form="unqualified"/>
  <xs:attribute name="ErrorMessagesList" type="xs:string"
                use="optional" default="" form="unqualified"/>
  <xs:attribute name="ErrorMessageLanguagesList" type="xs:string"
                use="optional" default="" form="unqualified"/>
  <xs:attribute name="IfObjectExists"
                type="DTS:IfObjectExistsEnum"
                use="optional" default="FailTask" form="unqualified"/>
  <xs:attribute name="SourceConnection" type="xs:string"
                use="optional" default="" form="unqualified"/>
  <xs:attribute name="DestinationConnection" type="xs:string"
                use="optional" default="" form="unqualified"/>
</xs:complexType>
```

The **TransferErrorMessagesTaskDataObjectDataType** complex type MUST NOT contain any elements or values.

The following table specifies the attributes of the
**TransferErrorMessagesTaskDataObjectDataType** complex type.

| Attribute | Description |
|---|---|
| TransferAllErrorMessages | A Boolean value that specifies whether to transfer all user-defined error messages or only the specified user-defined error messages. |
| | TRUE specifies that all user-defined error messages are transferred. |
| | FALSE specifies that only the specified user-defined messages are transferred. |
| ErrorMessageList | A string value that specifies a list of the error messages that are transferred. |
| ErrorMessageLanguagesList | A string value that specifies a list of languages for which the error messages are transferred. |
| IfObjectExists | An enumeration value that specifies what action is taken if an error message with the same name already exists at the destination. |
| SourceConnection | A string value that specifies a reference to a connection manager by its **DTSID** property value or by its **ObjectName** property value. |
| DestinationConnection | A string value that specifies a reference to a connection manager by its **DTSID** property value or by its **ObjectName** property value. |

### 2.7.1.19 TransferJobsTaskDataObjectDataType

The **TransferJobsTaskDataObjectDataType** complex type is used to specify parameters, properties, and settings that are specific to the Transfer Jobs Task.

The following is the XSD for the **TransferJobsTaskDataObjectDataType** complex type.

```
<xs:complexType name="TransferJobsTaskDataObjectDataType">
  <xs:sequence/>
  <xs:attribute name="TransferAllJobs" type="xs:boolean"
                use="optional" default="false" form="unqualified"/>
  <xs:attribute name="JobsList" type="xs:string"
                use="optional" default="" form="unqualified"/>
  <xs:attribute name="IfObjectExists" type="DTS:IfObjectExistsEnum"
                use="optional" default="FailTask" form="unqualified"/>
  <xs:attribute name="EnableJobsAtDestination" type="xs:boolean"
                use="optional" default="false" form="unqualified"/>
  <xs:attribute name="SourceConnection" type="xs:string"
                use="optional" default="" form="unqualified"/>
  <xs:attribute name="DestinationConnection" type="xs:string"
                use="optional" default="" form="unqualified"/>
</xs:complexType>
```

The **TransferJobsTaskDataObjectDataType** complex type MUST NOT contain any elements or values.

The following table specifies the attributes of the **TransferJobsTaskDataObjectDataType** complex type.

| Attribute | Description |
|---|---|
| TransferAllJobs | A Boolean value that specifies whether to transfer all jobs or only the specified jobs. |
| | TRUE specifies that all jobs MUST be transferred. |
| | FALSE specifies that only the specified jobs are transferred. |

| Attribute | Description |
|---|---|
| JobsList | A string value that specifies a list of jobs that are transferred. |
| IfObjectExists | An enumeration value that specifies what action is taken if a job of the same name already exists at the destination. |
| EnableJobsAtDestination | A Boolean value that specifies whether to enable jobs that are created at the destination.<br><br>TRUE specifies that jobs that are created at the destination MUST be enabled.<br><br>FALSE specifies that jobs that are created at the destination MUST NOT be enabled. |
| SourceConnection | A string value that specifies a reference to a connection manager by its **DTSID** property value or by its **ObjectName** property value. |
| DestinationConnection | A string value that specifies a reference to a connection manager by its **DTSID** property value or by its **ObjectName** property value. |

### 2.7.1.20    TransferLoginsTaskDataObjectDataType

The **TransferLoginsTaskDataObjectDataType** complex type is used to specify parameters, properties, and settings that are specific to the Transfer Logins Task.

The following is the XSD for the **TransferLoginsTaskDataObjectDataType** complex type.

```
<xs:complexType name="TransferLoginsTaskDataObjectDataType">
  <xs:sequence/>
  <xs:attribute name="LoginsToTransfer"
                type="DTS:TransferLoginsTaskLoginsToTransferEnum"
                use="optional" default="SelectedLogins"
                form="unqualified"/>
  <xs:attribute name="CopySids" type="xs:boolean" form="unqualified"
                use="optional" default="false"/>
  <xs:attribute name="LoginList" type="xs:string" form="unqualified"
                use="optional" default=""/>
  <xs:attribute name="IfObjectExists" type="DTS:IfObjectExistsEnum"
                use="optional" default="FailTask" form="unqualified"/>
  <xs:attribute name="SourceConnection" type="xs:string"
                use="optional" default="" form="unqualified"/>
  <xs:attribute name="DestinationConnection" type="xs:string"
                use="optional" default="" form="unqualified"/>
  <xs:attribute name="DatabasesList" type="xs:string"
                use="optional" default="" form="unqualified"/>
</xs:complexType>
```

The **TransferLoginsTaskDataObjectDataType** complex type MUST NOT contain any elements or values.

The following table specifies the attributes of the **TransferLoginsTaskDataObjectDataType** complex type.

| Attribute | Description |
|---|---|
| LoginsToTransfer | An enumeration value that specifies which logins MUST be transferred by the task. |
| CopySids | A Boolean value that specifies whether the security identifiers (SIDS) that are associated with logins are also transferred.<br><br>TRUE specifies that the SIDS MUST be transferred with the logins. |

| Attribute | Description |
|---|---|
|  | FALSE specifies that the SIDS MUST NOT be transferred with the logins. |
| LoginList | A string value that specifies the list of logins to be transferred. |
| IfObjectExists | An enumeration value that specifies what action is taken if a login of the same name already exists at the destination. |
| SourceConnection | A string value that specifies a reference to a connection manager by its **DTSID** property value or by its **ObjectName** property value. |
| DestinationConnection | A string value that specifies a reference to a connection manager by its **DTSID** property value or by its **ObjectName** property value. |
| DatabaseList | A string value that specifies the list of databases that exist on the source connection to transfer. |

#### 2.7.1.20.1 TransferLoginsTaskLoginsToTransferEnum

The **TransferLoginsTaskLoginsToTransferEnum** type contains the enumeration values for the **LoginsToTransfer** attribute.

The following is the XSD of the **TransferLoginsTaskLoginsToTransferEnum** type.

```
<xs:simpleType name="TransferLoginsTaskLoginsToTransferEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="SelectedLogins"/>
    <xs:enumeration value="AllLoginsFromSelectedDatabases"/>
    <xs:enumeration value="AllLogins"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **TransferLoginsTaskLoginsToTransferEnum** type.

| Enumeration value | Description |
|---|---|
| SelectedLogins | Specifies that the selected logins are transferred. |
| AllLoginsFromSelectedDatabases | Specifies that all logins from the selected databases are transferred. |
| AllLogins | Specifies that all logins are transferred. |

#### 2.7.1.21 TransferStoredProceduresTaskDataObjectDataType

The **TransferStoredProceduresTaskDataObjectDataType** complex type is used to specify parameters, properties, and settings that are specific to the Transfer Stored Procedures Task.

The following is the XSD for the **TransferStoredProceduresTaskDataObjectDataType** complex type.

```
<xs:complexType name="TransferStoredProceduresTaskDataObjectDataType">
  <xs:sequence/>
  <xs:attribute name="TransferAllStoredProcedures" type="xs:boolean" use="optional"
                default="false" form="unqualified"/>
```

```
                <xs:attribute name="StoredProceduresList" type="xs:string"
                              use="optional" default="" form="unqualified"/>
                <xs:attribute name="IfObjectExists" type="DTS:IfObjectExistsEnum"
                              use="optional" default="FailTask" form="unqualified"/>
                <xs:attribute name="SourceConnection" type="xs:string"
                              use="optional" default="" form="unqualified"/>
                <xs:attribute name="m_DestinationConnectionID" type="xs:string"
                              use="optional" default="" form="unqualified"/>
            </xs:complexType>
```

The **TransferStoredProceduresTaskDataObjectDataType** complex type MUST NOT contain any elements or values.

The following table specifies the attributes of the
**TransferStoredProceduresTaskDataObjectDataType** complex type.

| Attribute | Description |
|---|---|
| TransferAllStoredProcedures | A Boolean value that specifies whether to transfer all user-defined stored procedures in the master database or just the specified stored procedures. <br><br> TRUE specifies that all user-defined stored procedures in the master database are transferred. <br><br> FALSE specifies that only the specified stored procedures in the database are transferred. |
| StoredProceduresList | A string value that specifies the list of stored procedures in the master database that are transferred. |
| IfObjectExists | An enumeration value that specifies what action is taken if a stored procedure of the same name already exists at the destination. |
| SourceConnection | A string value that specifies a reference to a connection manager by its **DTSID** property value or by its **ObjectName** property value. |
| m_DestinationConnectionID | A string value that specifies a reference to a connection manager by its **DTSID** property value or by its **ObjectName** property value. |

### 2.7.1.22    TransferSqlServerObjectsTaskDataObjectDataType

The **TransferSqlServerObjectsTaskDataObjectDataType** complex type is used to specify
parameters, properties, and settings that are specific to the Transfer SQL Server Objects Task.

The following is the XSD for the **TransferSqlServerObjectsTaskDataObjectDataType** complex
type.

```
        <xs:complexType name="TransferSqlServerObjectsTaskDataObjectDataType">
          <xs:sequence/>
          <xs:attribute name="SourceConnection" type="xs:string"
                        use="optional" default="" form="unqualified"/>
          <xs:attribute name="DestinationConnection" type="xs:string"
                        use="optional" default="" form="unqualified"/>
          <xs:attribute name="SourceDatabase" type="xs:string"
                        use="optional" default="" form="unqualified"/>
          <xs:attribute name="DestinationDatabase" type="xs:string"
                        use="optional" default="" form="unqualified"/>
          <xs:attribute name="CopyAllObjects" type="xs:boolean"
                        use="optional" default="false" form="unqualified"/>
          <xs:attribute name="CopyAllRules" type="xs:boolean"
                        use="optional" default="false" form="unqualified"/>
          <xs:attribute name="CopyAllTables" type="xs:boolean"
```

```
                        use="optional" default="false" form="unqualified"/>
            <xs:attribute name="CopyAllViews" type="xs:boolean"
                        use="optional" default="false" form="unqualified"/>
            <xs:attribute name="CopyAllStoredProcedures"
                        type="xs:boolean" use="optional" default="false"
                        form="unqualified"/>
            <xs:attribute name="CopyAllUserDefinedFunctions"
                        type="xs:boolean" use="optional" default="false"
                        form="unqualified"/>
            <xs:attribute name="CopyAllDefaults" type="xs:boolean"
                        use="optional" default="false" form="unqualified"/>
            <xs:attribute name="CopyAllUserDefinedDataTypes"
                        use="optional" default="false" type="xs:boolean"
                        form="unqualified"/>
            <xs:attribute name="CopyAllPartitionFunctions" type="xs:boolean"
                        use="optional" default="false" form="unqualified"/>
            <xs:attribute name="CopyAllPartitionSchemes" type="xs:boolean"
                        use="optional" default="false" form="unqualified"/>
            <xs:attribute name="CopyAllSchemas" type="xs:boolean"
                        use="optional" default="false" form="unqualified"/>
            <xs:attribute name="CopyAllSqlAssemblies" type="xs:boolean"
                        use="optional" default="false" form="unqualified"/>
            <xs:attribute name="CopyAllUserDefinedAggregates"
                        use="optional" default="false" type="xs:boolean"
                        form="unqualified"/>
            <xs:attribute name="CopyAllUserDefinedTypes" type="xs:boolean"
                        use="optional" default="false" form="unqualified"/>
            <xs:attribute name="CopyAllXmlSchemaCollections" type="xs:boolean"
                        use="optional" default="false" form="unqualified"/>
            <xs:attribute name="LoginsList" type="xs:string"
                        use="optional" default="" form="unqualified"/>
            <xs:attribute name="UsersList" type="xs:string"
                        use="optional" default="" form="unqualified"/>
            <xs:attribute name="RulesList" type="xs:string"
                        use="optional" default="" form="unqualified"/>
            <xs:attribute name="TablesList" type="xs:string"
                        use="optional" default="" form="unqualified"/>
            <xs:attribute name="ViewsList" type="xs:string"
                        use="optional" default="" form="unqualified"/>
            <xs:attribute name="StoredProceduresList"
                        type="xs:string" use="optional" default=""
                        form="unqualified"/>
            <xs:attribute name="UserDefinedFunctionsList"
                        type="xs:string" use="optional" default=""
                        form="unqualified"/>
            <xs:attribute name="DefaultsList" type="xs:string"
                        use="optional" default="" form="unqualified"/>
            <xs:attribute name="UserDefinedDataTypesList"
                        type="xs:string" use="optional" default=""
                        form="unqualified"/>
            <xs:attribute name="PartitionFunctionsList" type="xs:string"
                        use="optional" default="" form="unqualified"/>
            <xs:attribute name="PartitionSchemesList" type="xs:string"
                        use="optional" default="" form="unqualified"/>
            <xs:attribute name="SchemasList" type="xs:string"
                        use="optional" default="" form="unqualified"/>
            <xs:attribute name="SqlAssembliesList" type="xs:string"
                        use="optional" default="" form="unqualified"/>
            <xs:attribute name="UserDefinedAggregatesList"
                        type="xs:string" use="optional" default=""
                        form="unqualified"/>
            <xs:attribute name="UserDefinedTypesList" type="xs:string"
                        use="optional" default="" form="unqualified"/>
            <xs:attribute name="XmlSchemaCollectionsList" type="xs:string"
                        use="optional" default="" form="unqualified"/>
            <xs:attribute name="DropObjectsFirst" type="xs:boolean"
                        use="optional" default="false" form="unqualified"/>
            <xs:attribute name="IncludeDependentObjects"
                        type="xs:boolean" use="optional" default="false"
                        form="unqualified"/>
```

```
        <xs:attribute name="IncludeExtendedProperties" type="xs:boolean"
                    use="optional" default="false" form="unqualified"/>
        <xs:attribute name="CopyData" type="xs:boolean"
                    use="optional" default="false" form="unqualified"/>
        <xs:attribute name="ExistingData"
            type="DTS:TransferSqlServerObjectsTaskExistingDataEnum"
                    use="optional" default="Replace" form="unqualified"/>
        <xs:attribute name="CopySchema" type="xs:boolean"
                    use="optional" default="true" form="unqualified"/>
        <xs:attribute name="UseCollation" type="xs:boolean"
                    use="optional" default="false" form="unqualified"/>
        <xs:attribute name="CopyDatabaseUsers" type="xs:boolean"
                    use="optional" default="false" form="unqualified"/>
        <xs:attribute name="CopyDatabaseRoles" type="xs:boolean"
                    use="optional" default="false" form="unqualified"/>
        <xs:attribute name="CopySqlServerLogins" type="xs:boolean"
                    use="optional" default="false" form="unqualified"/>
        <xs:attribute name="CopyObjectLevelPermissions" type="xs:boolean"
                    use="optional" default="false" form="unqualified"/>
        <xs:attribute name="CopyIndexes" type="xs:boolean"
                    use="optional" default="false" form="unqualified"/>
        <xs:attribute name="CopyTriggers" type="xs:boolean"
                    use="optional" default="false" form="unqualified"/>
        <xs:attribute name="CopyFullTextIndexes" type="xs:boolean"
                    use="optional" default="false" form="unqualified"/>
        <xs:attribute name="CopyPrimaryKeys" type="xs:boolean"
                    use="optional" default="false" form="unqualified"/>
        <xs:attribute name="CopyForeignKeys" type="xs:boolean"
                    use="optional" default="false" form="unqualified"/>
        <xs:attribute name="GenerateScriptsInUnicode" type="xs:boolean"
                    use="optional" default="false" form="unqualified"/>
        <xs:attribute name="SourceTranslateChar" type="xs:boolean"
                    use="optional" default="false" form="unqualified"/>
        <xs:attribute name="DestinationTranslateChar" type="xs:boolean"
                    use="optional" default="false" form="unqualified"/>
        <xs:attribute name="CopyAllDRIObjects" type="xs:boolean"
                    use="optional" default="false" form="unqualified"/>
    </xs:complexType>
```

The **TransferSqlServerObjectsTaskDataObjectDataType** complex type MUST NOT contain any elements or values.

The following table specifies the attributes of the **TransferSqlServerObjectsTaskDataObjectDataType** complex type.

| Attribute | Description |
|---|---|
| SourceConnection | A string value that specifies a reference to the connection manager for the source of the data to be transferred. The connection manager can be referenced by its **DTSID** property or by its **ObjectName** property. |
| DestinationConnection | A string value that specifies a reference to the connection manager for the destination of the data to be transferred. The connection manager can be referenced by its **DTSID** property or by its **ObjectName** property. |
| SourceDatabase | A string value that specifies the source database for objects to be transferred. |
| DestinationDatabase | A string value that specifies the destination database for objects to be transferred. |
| CopyAllObjects | A Boolean value that specifies whether the task copies all objects or only the specified objects. TRUE specifies that all objects from the source database are copied. FALSE specifies that only the specified objects are copied. |

| Attribute | Description |
|---|---|
| CopyAllRules | A Boolean value that specifies whether the task copies all rule objects or only the specified rule objects.<br><br>TRUE specifies that all rule objects from the source database are copied.<br><br>FALSE specifies that only the rules that are specified in the **RulesList** attribute are copied. |
| CopyAllTables | A Boolean value that specifies whether the task copies all tables or only the specified tables.<br><br>TRUE specifies that all tables from the source database are copied.<br><br>FALSE specifies that only the tables that are specified in the **TablesList** attribute are copied. |
| CopyAllViews | A Boolean value that specifies whether the task copies all views or only the specified views.<br><br>TRUE specifies that all views from the source database are copied.<br><br>FALSE specifies that only the views that are specified in the **ViewsList** attribute are copied. |
| CopyAllStoredProcedures | A Boolean value that specifies whether the task copies all stored procedures or only the specified stored procedures.<br><br>TRUE specifies that all stored procedures from the source database are copied.<br><br>FALSE specifies that only the stored procedures specified in the **StoredProceduresList** attribute are copied. |
| CopyAllUserDefinedFunctions | A Boolean value that specifies whether the task copies all user-defined functions or only the specified user-defined functions.<br><br>TRUE specifies that all user-defined functions from the source database are copied.<br><br>FALSE specifies that only the user-defined functions that are specified in the **UserDefinedFunctionsList** attribute are copied. |
| CopyAllDefaults | A Boolean value that specifies whether the task copies all defaults or only the specified defaults.<br><br>TRUE specifies that all defaults from the source database are copied.<br><br>FALSE specifies that only the defaults specified in the **DefaultsList** attribute are copied. |
| CopyAllUserDefinedDataTypes | A Boolean value that specifies whether the task copies all user-defined data types or only the specified user-defined data types.<br><br>TRUE specifies that all user-defined data types from the source database are copied.<br><br>FALSE specifies that only the user-defined data types specified in the **UserDefinedTypesList** attribute are copied. |
| CopyAllPartitionFunctions | A Boolean value that specifies whether the task copies all partition functions or only the specified partition functions.<br><br>TRUE specifies that all partition functions from the source database are copied.<br><br>FALSE specifies that only the defaults specified in the **PartitionFunctionsList** attribute are copied. |
| CopyAllPartitionSchemas | A Boolean value that specifies whether the task copies all partition schemas or only the specified partition schemas.<br><br>TRUE specifies that all partition schemas from the source database are copied.<br><br>FALSE specifies that only the partition schemas specified in the **PartitionSchemesList** attribute are copied. |
| CopyAllSchemas | A Boolean value that specifies whether the task copies all schemas or only the |

| Attribute | Description |
|---|---|
|  | specified schemas. |
|  | TRUE specifies that all schemas from the source database are copied. |
|  | FALSE specifies that only the schemas specified in the **SchemasList** attribute are copied. |
| CopyAllSqlAssemblies | A Boolean value that specifies whether the task copies all SQL assemblies or only the specified SQL assemblies. |
|  | TRUE specifies that all SQL assemblies from the source database are copied. |
|  | FALSE specifies that only the SQL assemblies specified in the **SqlAssembliesList** attribute are copied. |
| CopyAllUserDefinedAggregates | A Boolean value that specifies whether the task copies all user-defined aggregates or only the specified user-defined aggregates. |
|  | TRUE specifies that all user-defined aggregates from the source database are copied. |
|  | FALSE specifies that only the user-defined aggregates specified in the **UserDefinedAggregatesList** attribute are copied. |
| CopyAllUserDefinedTypes | A Boolean value that specifies whether the task copies all user-defined types or only the specified user-defined types. |
|  | TRUE specifies that all user-defined types from the source database are copied. |
|  | FALSE specifies that only the user-defined types that are specified in the **UserDefinedTypesList** attribute are copied. |
| CopyAllXmlSchemaCollections | A Boolean value that specifies whether the task copies all XML schema collections or only the specified XML schema collections. |
|  | TRUE specifies that all XML schema collections from the source database are copied. |
|  | FALSE specifies that only the XML schema collections that are specified in the **XmlSchemaCollectionsList** attribute are copied. |
| LoginsList | A string value that specifies the list of logins to copy. |
| UsersList | A string value that specifies the list of database users to copy. |
| RulesList | A string value that specifies the list of rules to copy. |
| TablesList | A string value that specifies the list of tables to copy. |
| ViewsList | A string value that specifies the list of views to copy. |
| StoredProceduresList | A string value that specifies the list of stored procedures to copy. |
| UserDefinedFunctionsList | A string value that specifies the list of user-defined functions to copy. |
| DefaultsList | A string value that specifies the list of defaults to copy. |
| UserDefinedDataTypesList | A string value that specifies the list of user-defined data types to copy. |
| PartitionFunctionsList | A string value that specifies the list of partition functions to copy. |
| PartitionSchemesList | A string value that specifies the list of partition schemas to copy. |
| SchemasList | A string value that specifies the list of schemas to copy. |
| SqlAssembliesList | A string value that specifies the list of SQL assemblies to copy. |
| UserDefinedAggregatesList | A string value that specifies the list of user-defined aggregates to copy. |
| UserDefinedTypesList | A string value that specifies the list of user-defined types to copy. |

| Attribute | Description |
|-----------|-------------|
| XmlSchemaCollectionsList | A string value that specifies the list of XML schema collections to copy. |
| DropObjectsFirst | A Boolean value that specifies whether to drop object on the destination database before the transfer. This option is used only if the value of the **CopySchema** attribute is set to TRUE.<br><br>TRUE specifies that objects on the destination server are dropped before the transfer.<br><br>FALSE specifies that objects on the destination server are not dropped before the transfer. |
| IncludeDependentObjects | A Boolean value that specifies whether dependent objects are included with the transfer.<br><br>TRUE specifies that dependent objects are included with the transfer.<br><br>FALSE specifies that dependent objects are not included with the transfer. |
| IncludeExtendedProperties | A Boolean value that specifies whether extended properties are included with the transfer.<br><br>TRUE specifies that extended properties are included with the transfer.<br><br>FALSE specifies that extended properties are not included with the transfer. |
| CopyData | A Boolean value that specifies whether data is included with the transfer.<br><br>TRUE specifies that data is included with the transfer.<br><br>FALSE specifies that data is not included with the transfer. |
| ExistingData | An enumeration value that specifies whether to append or replace existing data. This value is used only if the **CopyData** attribute is set to TRUE and the **DropObjectsFirst** attribute is set to FALSE. |
| CopySchema | A Boolean value that specifies whether the schema is included with the transfer.<br><br>TRUE specifies that the schema is included with the transfer.<br><br>FALSE specifies that the schema is not included with the transfer. |
| UseCollation | A Boolean value that specifies whether the transfer includes a collation.<br><br>TRUE specifies that a collation is included with the transfer.<br><br>FALSE specifies that a collation is not included with the transfer. |
| CopyDatabaseUsers | A Boolean value that specifies whether database users are included in the transfer.<br><br>TRUE specifies that database users are included with the transfer. The database users that are specified in the **UsersList** attribute are copied.<br><br>FALSE specifies that database users are not included with the transfer. |
| CopyDatabaseRoles | A Boolean value that specifies whether database roles are included in the transfer.<br><br>TRUE specifies that database roles are included with the transfer.<br><br>FALSE specifies that database roles are not included with the transfer. |
| CopySqlServerLogins | A Boolean value that specifies whether database logins[103] are included in the transfer.<br><br>TRUE specifies that database logins are included with the transfer.<br><br>FALSE specifies that database logins are not included with the transfer. |
| CopyObjectLevelPermissions | A Boolean value that specifies whether object-level permissions are included in the transfer.<br><br>TRUE specifies that object-level permissions are included with the transfer.<br><br>FALSE specifies that object level permissions are not included with the transfer. |

| Attribute | Description |
|---|---|
| CopyIndexes | A Boolean value that specifies whether indexes are included in the transfer.<br>TRUE specifies that indexes are included in the transfer.<br>FALSE specifies that indexes are not included in the transfer. |
| CopyTriggers | A Boolean value that specifies whether triggers are included in the transfer.<br>TRUE specifies that triggers are included in the transfer.<br>FALSE specifies that triggers are not included in the transfer. |
| CopyFullTextIndexes | A Boolean value that specifies whether indexes are included in the transfer.<br>TRUE specifies that triggers are included in the transfer.<br>FALSE specifies that triggers are not included in the transfer. |
| CopyPrimaryKeys | A Boolean value that specifies whether primary key constraints are included in the transfer.<br>TRUE specifies that primary key constraints are included in the transfer.<br>FALSE specifies that primary key constraints are not included in the transfer. |
| CopyForeignKeys | A Boolean value that specifies whether foreign key constraints are included in the transfer.<br>TRUE specifies that foreign key constraints are included in the transfer.<br>FALSE specifies that foreign key constraints are not included in the transfer. |
| GenerateScriptsInUnicode | A Boolean value that specifies whether the script MUST be generated in Unicode format.<br>TRUE specifies that the script MUST be generated in Unicode format.<br>FALSE specifies that the script MUST NOT be generated in Unicode format. |
| SourceTranslateChar | A Boolean value that specifies whether the transfer operation translates the codepage for characters on the source database.<br>TRUE specifies that the transfer operation translates the codepage for characters on the source database.<br>FALSE specifies that the transfer operation does not translate the codepage for characters on the source database. |
| DestinationTranslateChar | A Boolean value that specifies whether the transfer operation translates the codepage for characters on the destination database.<br>TRUE specifies that the transfer operation translates the codepage for characters on the destination database.<br>FALSE specifies that the transfer operation does not translate the codepage for characters on the destination database. |
| CopyAllDRIObjects | A Boolean value that specifies that data referential integrity (DRI) objects are included in the transfer.<br>TRUE specifies that DRI objects are included in the transfer.<br>FALSE specifies that DRI objects are not included in the transfer. |

### 2.7.1.22.1    TransferSqlServerObjectsTaskExistingDataEnum

The **TransferSqlServerObjectsTaskExistingDataEnum** type contains the enumeration values for the **LoginsToTransfer** attribute.

The following is the XSD of the **TransferSqlServerObjectsTaskExistingDataEnum** type.

```
<xs:simpleType name="TransferSqlServerObjectsTaskExistingDataEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Replace"/>
    <xs:enumeration value="Append"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the
**TransferSqlServerObjectsTaskExistingDataEnum** type.

| Enumeration value | Description |
|---|---|
| Replace | Specifies that new data replaces existing data. |
| Append | Specifies that new data is appended to existing data. |

### 2.7.1.23 WebServiceTaskData

The **WebServiceTaskData** element is used to specify parameters, properties, and settings that are
specific to the execution of a Web Service Task. This element resides in the WebServiceTask
namespace. The ExecutableObjectDataType type refers to this element within an XSD **choice**
element, using the XSD **ref** attribute, in the following XSD fragment.

```
<xs:element ref="WSTask:WebServiceTaskData"/>
```

### 2.7.1.23.1 WebServiceTask Namespace

The **WebServiceTask** namespace specifies the elements and types that are used by the Web Service
Task.

The following is the XSD declaration for the **WebServiceTask** namespace.

```
xmlns:WSTask="www.microsoft.com/sqlserver/dts/tasks/webservicetask"
```

### 2.7.1.23.1.1 WebServiceTaskData

The following is the XSD for the element declaration for the **WebServiceTaskData** element in the
WebServiceTask namespace.

```
<xs:element name="WebServiceTaskData" type="WSTask:WSTaskDataType"/>
```

### 2.7.1.23.1.1.1 WSTaskDataType

The **WSTaskDataType** complex type holds parameters, properties, and settings for the Web Service
Task.

The following is the XSD for the **WSTaskDataType** complex type.

```
<xs:complexType name="WSTaskDataType">
  <xs:sequence>
    <xs:element name="MethodInfo" type="WSTask:WebServiceTaskDataMethodInfoType"/>
  </xs:sequence>
  <xs:attribute name="ConnectionName" type="xs:string" use="optional" default=""/>
```

```
        <xs:attribute name="Service" type="xs:string" use="optional" default=""/>
        <xs:attribute name="WSDLFile" type="xs:string" use="optional" default=""/>
        <xs:attribute name="OverwriteWSDLFile" use="optional" default="False">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="True"/>
              <xs:enumeration value="False"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="OutputType" type="WSTask:WebServiceTaskDataOutputTypeEnum"
                      use="optional" default="File"/>
        <xs:attribute name="OutputLocation" type="xs:string" use="optional" default=""/>
      </xs:complexType>
```

The following table provides additional information about the elements and types for the
**WSTaskDataType** complex type.

| Element | Type definition | Description |
|---------|-----------------|-------------|
| MethodInfo | WebServiceTaskDataMethodInfoType | A complex type that specifies the method that is executed by the Web Service Task. |

The following table specifies the attributes of the **WSTaskDataType** complex type.

| Attribute | Description |
|-----------|-------------|
| ConnectionName | A string value that specifies the name of the connection that is used to execute the Web Service Task. |
| Service | A string value that specifies the service name of the web service to be executed. |
| WSDLFile | A string value that specifies the name of the file that contains the Web Services Description Language (WSDL) file that is used with the web service. |
| OverwriteWSDLFile | An enumeration value that specifies whether to overwrite the WSDL file.<br>"True" specifies that the WSDL file MUST be overwritten.<br>"False" specifies that the WSDL file MUST NOT be overwritten. |
| OutputType | An enumeration value that specifies the destination for the output. |
| OutputLocation | A string values that specifies the output location. Depending on the value of the **OutputType** attribute, this string value contains either a variable name or a file connection reference. |

#### 2.7.1.23.1.1.1.1    WebServiceTaskDataMethodInfoType

The **WebServiceTaskDataMethodInfoType** complex type is used to specify information about the
method that is called by the Web Service Task.

The following is the XSD for the **WebServiceTaskDataMethodInfoType** complex type.

```
      <xs:complexType name="WebServiceTaskDataMethodInfoType">
        <xs:sequence>
          <xs:element name="Documentation" type="xs:string" minOccurs="0" maxOccurs="1"/>
          <xs:element name="ParamInfo" type="WSTask:WebServiceTaskDataParamInfoType"
                      maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="MethodName" type="xs:string"/>
```

```
                <xs:attribute name="MessageName" type="xs:string"/>
        </xs:complexType>
```

The following table provides additional information about the elements and types for the **WebServiceTaskDataMethodInfoType** complex type.

| Element | Type definition | Description |
|---------|-----------------|-------------|
| Documentation | xs:string | A string value that contains descriptive documentation for the **MethodInfo** element contents. |
| ParamInfo | WebServiceTaskDataParamInfoType | A complex type that contains information about the parameters that the called method uses. |

The following table specifies additional information for the attributes of the **WebServiceTaskDataMethodInfoType** complex type.

| Attribute | Description |
|-----------|-------------|
| MethodName | A string value that specifies the name of the method that is executed by the Web Service Task. |
| MessageName | A string value that specifies the name of the message that the web service is accessing. |

#### 2.7.1.23.1.1.1.2    WebServiceTaskDataParamInfoType

The **WebServiceTaskDataParamInfoType** complex type is used to specify information about the method that is called by the Web Service Task.

The following is the XSD for the **WebServiceTaskDataParamInfoType** complex type.

```
    <xs:complexType name="WebServiceTaskDataParamInfoType">
      <xs:sequence>
        <xs:element name="VariableValue" type="xs:string" minOccurs="0" maxOccurs="1"/>
        <xs:element name="PrimitiveValue" type="xs:anySimpleType" minOccurs="0"
                    maxOccurs="1"/>
      </xs:sequence>
      <xs:attribute name="Name" type="xs:string"/>
      <xs:attribute name="Datatype" type="xs:string"/>
      <xs:attribute name="ParamType" type="WSTask:ParamTypeEnum"/>
      <xs:attribute name="SeqNumber" type="xs:int"/>
    </xs:complexType>
```

The following table provides additional information about the elements and types for the **WebServiceTaskDataParamInfoType** complex type.

| Element | Type definition | Description |
|---------|-----------------|-------------|
| VariableValue | xs:string | A string value that specifies the name of the variable that supplies the value of the parameter. This element appears only if the value of the **ParamType** attribute is equal to "Variable". |
| PrimitiveValue | xs:anySimpleType | A value of anySimpleType that contains the value of the parameter. This element appears only if the value of the **ParamType** attribute is equal to "Primitive". |

The following table specifies additional information for the attributes of the **WebServiceTaskDataParamInfoType** complex type.

| Attribute | Description |
|---|---|
| Name | A string value that specifies the name of the parameter. |
| Datatype | A string value that specifies the data type of the parameter. The strings that can be used for the value of the **Datatype** attribute are defined in the WSDL file of the web service that is being called. |
| ParamType | An enumeration value that specifies the type of the parameter. |
| SeqNumber | An integer value that specifies the sequence number of the parameter. |

### 2.7.1.23.1.1.1.3      WebServiceTaskDataOutputTypeEnum

The **WebServiceTaskDataOutputTypeEnum** type contains the enumeration values for the **OutputType** attribute.

The following is the XSD of the **WebServiceTaskDataOutputTypeEnum** type.

```
<xs:simpleType name="WebServiceTaskDataOutputTypeEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="File"/>
    <xs:enumeration value="Variable"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **WebServiceTaskDataOutputTypeEnum** type.

| Enumeration value | Description |
|---|---|
| File | Specifies that the output is directed to a file. |
| Variable | Specifies that the output is directed to a variable. |

### 2.7.1.24      WMIDRTaskDataObjectDataType

The **WMIDRTaskDataObjectDataType** complex type is used to specify parameters, properties, and settings that are specific to the WMI Data Reader Task.

The following is the XSD for the **WMIDRTaskDataObjectDataType** complex type.

```
<xs:complexType name="WMIDRTaskDataObjectDataType">
  <xs:sequence/>
  <xs:attribute name="WQLQuerySourceType" type="DTS:SourceTypeEnum"
                use="optional" default="DirectInput" form="unqualified"/>
  <xs:attribute name="WQLQuerySource" type="xs:string"
                use="optional" default="" form="unqualified"/>
  <xs:attribute name="DestinationType"
                type="DTS:DestinationTypeEnum" form="unqualified"
                use="optional" default="FileConnection"/>
  <xs:attribute name="Destination" type="xs:string"
                use="optional" default="" form="unqualified"/>
  <xs:attribute name="OverwriteDestination"
                type="DTS:WMITaskDataOverwriteDestinationEnum"
                use="optional" default="KeepOriginal" form="unqualified"/>
  <xs:attribute name="OutputType"
```

```
                  type="DTS:WMITaskDataOutputTypeEnum"
                  use="optional" default="DataTable" form="unqualified"/>
    <xs:attribute name="WMIConnectionName" type="xs:string"
                  use="optional" default="" form="unqualified"/>
</xs:complexType>
```

The **WMIDRTaskDataObjectDataType** complex type MUST NOT contain any elements or values.

The following table specifies the attributes of the **WMIDRTaskDataObjectDataType** complex type.

| Attribute | Description |
|-----------|-------------|
| WQLQuerySourceType | An enumeration value that specifies where the source for the WMI Query Language (WQL) is found. |
| WQLQuerySource | A string value that specifies the source of the WQL query. This value is used only if the **WQLQuerySourceType** value is set to "DirectInput". |
| DestinationType | An enumeration value that specifies the type of the destination where results are sent. |
| Destination | A string value that specifies the destination where results are sent. This attribute contains a variable name or a connection manager reference, depending on the value of the **DestinationType** attribute. |
| OverwriteDestination | An enumeration value that specifies how output that already exists at the destination is handled. |
| OutputType | An enumeration value that specifies the type of values that are output and the way in which the output values are formatted. |
| WMIConnectionName | A string value that specifies the connection name. The connection name is a reference to a connection manager, and the connection name can match either the **DTSID** property of the connection manager or the **ObjectName** property of the connection manager. |

### 2.7.1.24.1   WMITaskDataOverwriteDestinationEnum

The **WMITaskDataOverwriteDestinationEnum** type contains the enumeration values for the **OverwriteDestination** attribute.

The following is the XSD of the **WMITaskDataOverwriteDestinationEnum** type.

```
<xs:simpleType name="WMITaskDataOverwriteDestinationEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="KeepOriginal"/>
    <xs:enumeration value="AppendToDestination"/>
    <xs:enumeration value="OverwriteDestination"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the
**WMITaskDataOverwriteDestinationEnum** type.

| Enumeration value | Description |
|-------------------|-------------|
| KeepOriginal | Specifies that the original data is kept and that no data is overwritten. |
| AppendToDestination | Specifies that new data from this task is appended to any existing data at the destination. |

| Enumeration value | Description |
|---|---|
| OverwriteDestination | Specifies that any existing destination data is overwritten with new data from the results of this task. |

### 2.7.1.24.2 WMITaskDataOutputTypeEnum

The **WMITaskDataOutputTypeEnum** type contains the enumeration values for the **OutputType** attribute.

The following is the XSD of the **WMITaskDataOutputTypeEnum** type.

```
<xs:simpleType name="WMITaskDataOutputTypeEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="DataTable"/>
    <xs:enumeration value="PropertyNameAndValue"/>
    <xs:enumeration value="PropertyValue"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **WMITaskDataOutputTypeEnum** type.

| Enumeration value | Description |
|---|---|
| DataTable | Specifies that the output is formatted as a data table. |
| PropertyNameAndValue | Specifies that the output is in the form of property name/property value pairs, with one pair on each line. |
| PropertyValue | Specifies that only property values are output, one per line. |

### 2.7.1.25 WMIEWTaskDataObjectDataType

The **WMIEWTaskDataObjectDataType** complex type is used to specify parameters, properties, and settings that are specific to the WMI Event Watcher Task.

The following is the XSD for the **WMIEWTaskDataObjectDataType** complex type.

```
<xs:complexType name="WMIEWTaskDataObjectDataType">
  <xs:sequence/>
  <xs:attribute name="WMIConnectionName" type="xs:string"
                use="optional" default="" form="unqualified"/>
  <xs:attribute name="WQLQuerySourceType" type="DTS:SourceTypeEnum"
                use="optional" default="DirectInput" form="unqualified"/>
  <xs:attribute name="WQLQuerySource" type="xs:string"
                use="optional" default="" form="unqualified"/>
  <xs:attribute name="ActionAtEvent"
                type="DTS:WMIEWTaskDataActionAtEventEnum"
                use="optional" default="LogTheEventAndFireDTSEvent"
                form="unqualified"/>
  <xs:attribute name="AfterEvent"
                type="DTS:WMIEWTaskDataAfterEventEnum"
                use="optional" default="ReturnWithSuccess"
                form="unqualified"/>
  <xs:attribute name="ActionAtTimeout"
                type="DTS:WMIEWTaskDataActionAtTimeoutEnum"
                use="optional" default="LogTimeoutAndFireDTSEvent"
```

```
                  form="unqualified"/>
    <xs:attribute name="AfterTimeout"
                  type="DTS:WMIEWTaskDataAfterEventEnum"
                  use="optional" default="ReturnWithFailure"
                  form="unqualified"/>
    <xs:attribute name="NumberOfEvents" type="xs:int"
                  use="optional" default="1" form="unqualified"/>
    <xs:attribute name="Timeout" type="xs:int" form="unqualified"
                  use="optional" default="0"/>
</xs:complexType>
```

The **WMIEWTaskDataObjectDataType** complex type MUST NOT contain any elements or values.

The following table specifies the attributes of the **WMIEWTaskDataObjectDataType** complex type.

| Attribute | Description |
|---|---|
| WMIConnectionName | A string value that specifies the connection name. The **WMIConnectionName** value is a reference to a connection manager; the connection name can match either the **DTSID** property of the connection manager or the **ObjectName** property of the connection manager. |
| WQLQuerySourceType | An enumeration value that specifies where the source for the WQL query is found. |
| WQLQuerySource | A string value that specifies the source of the WQL query. This value is used only if the **WQLQuerySourceType** value is set to "DirectInput". |
| ActionAtEvent | An enumeration value that specifies the action to take when the event occurs. |
| AfterEvent | An enumeration value that specifies the action to take after the event occurs. |
| ActionAtTimeout | An enumeration value that specifies the action to take when timeout occurs. |
| AfterTimeout | An enumeration value that specifies the action to take after timeout occurs. |
| NumberOfEvents | An integer value that specifies the number of events that the task watches. |
| Timeout | An integer value that specifies the timeout interval for the task, in seconds. |

#### 2.7.1.25.1   WMIEWTaskDataActionAtEventEnum

The **WMIEWTaskDataActionAtEventEnum** type contains the enumeration values for the **ActionAtEvent** attribute and the **ActionAtTimeout** attribute.

The following is the XSD of the **WMIEWTaskDataActionAtEventEnum** type.

```
<xs:simpleType name="WMIEWTaskDataActionAtEventEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="LogTheEventAndFireDTSEvent"/>
    <xs:enumeration value="LogTheEvent"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **WMIEWTaskDataActionAtEventEnum** type.

| Enumeration value | Description |
|---|---|
| LogTheEventAndFireDTSEvent | Specifies that the event is logged and a DTS event is fired. |

| Enumeration value | Description |
| --- | --- |
| LogTheEvent | Specifies that the event is logged only. |

### 2.7.1.25.2    WMIEWTaskDataAfterEventEnum

The **WMIEWTaskDataAfterEventEnum** type contains the enumeration values for the **ActionAfterEvent** attribute.

The following is the XSD of the **WMIEWTaskDataAfterEventEnum** type.

```
<xs:simpleType name="WMIEWTaskDataAfterEventEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="WatchfortheEventAgain"/>
    <xs:enumeration value="ReturnWithSuccess"/>
    <xs:enumeration value="ReturnWithFailure"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **WMIEWTaskDataAfterEventEnum** type.

| Enumeration value | Description |
| --- | --- |
| WatchfortheEventAgain | Specifies that the event is watched for again. |
| ReturnWithSuccess | Specifies that a success exit value is returned. |
| ReturnWithFailure | Specifies that a failure exit value is returned. |

### 2.7.1.25.3    WMIEWTaskDataActionAtTimeoutEnum

The **WMIEWTaskDataActionAtTimeoutEnum** type contains the enumeration values for the **ActionAtTimeout** attribute.

The following is the XSD of the **WMIEWTaskDataActionAtTimeoutEnum** type.

```
<xs:simpleType name="WMIEWTaskDataActionAtTimeoutEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="LogTimeoutAndFireDTSEvent"/>
    <xs:enumeration value="LogTimeout"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **WMIEWTaskDataActionAtTimeoutEnum** type.

| Enumeration value | Description |
| --- | --- |
| LogTimeoutAndFireDTSEvent | Specifies that a timeout event is logged and a DTS event is fired. |
| LogTimeout | Specifies that a timeout event is logged only. |

### 2.7.1.26    XMLTaskDataObjectDataType

The **XMLTaskDataObjectDataType** complex type is used to specify parameters, properties, and settings that are specific to the XML Task.

The following is the XSD for the **XMLTaskDataObjectDataType** complex type.

```
<xs:complexType name="XMLTaskDataObjectDataType">
  <xs:sequence>
  </xs:sequence>
  <xs:attribute name="OperationType" form="unqualified"
                type="DTS:XMLTaskOperationTypeEnum"
                use="optional" default="Diff"/>
  <xs:attribute name="SourceType" type="DTS:SourceTypeEnum"
                use="optional" default="FileConnection"
                form="unqualified"/>
  <xs:attribute name="Source" type="xs:string" form="unqualified"
                use="optional" default="" />
  <xs:attribute name="SecondOperandType" type="DTS:SourceTypeEnum"
                use="optional" default="DirectInput" form="unqualified"/>
  <xs:attribute name="SecondOperand" type="xs:string"
                use="optional" default="" form="unqualified"/>
  <xs:attribute name="SaveOperationResult" form="unqualified"
                use="optional" default="False">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="True"/>
        <xs:enumeration value="False"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="DestinationType" type="DTS:DestinationTypeEnum"
                form="unqualified" use="optional" default="FileConnection" />
  <xs:attribute name="Destination" type="xs:string"
                use="optional" default="" form="unqualified"/>
  <xs:attribute name="OverwriteDestination" form="unqualified"
                use="optional" default="False">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="True"/>
        <xs:enumeration value="False"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="FailOpOnValidationFail" form="unqualified"
                use="optional" default="False">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="True"/>
        <xs:enumeration value="False"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="FailOpOnDifference" form="unqualified"
                use="optional" default="False">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="True"/>
        <xs:enumeration value="False"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="SaveDiffGram" form="unqualified"
                use="optional" default="False">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="True"/>
        <xs:enumeration value="False"/>
```

```
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="DiffGramDestinationType" form="unqualified"
                  type="DTS:DestinationTypeEnum" use="optional"
                  default="FileConnection"/>
    <xs:attribute name="DiffGramDestination" form="unqualified"
                  type="xs:string" use="optional" default=""/>
    <xs:attribute name="DiffOptions" type="DTS:XMLTaskDiffOptionsType"
                  use="optional" default="0" form="unqualified"/>
    <xs:attribute name="DiffAlgorithm" form="unqualified"
                  type="DTS:XMLTaskDiffAlgorithmEnum"
                  use="optional" default="Auto"/>
    <xs:attribute name="XPathStringSourceType"
                  type="DTS:SourceTypeEnum" form="unqualified"
                  use="optional" default="DirectInput"/>
    <xs:attribute name="XPathStringSource" type="xs:string"
                  use="optional" default="" form="unqualified"/>
    <xs:attribute name="XPathOperation" form="unqualified"
                  type="DTS:XMLTaskXPathOperationEnum"
                  use="optional" default="Evaluation" />
    <xs:attribute name="PutResultInOneNode" form="unqualified"
                  use="optional" default="False">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="True"/>
          <xs:enumeration value="False"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="ValidationType" type="DTS:XMLTaskValidationTypeEnum"
                  use="optional" default="XSD" form="unqualified"/>
  </xs:complexType>
```

The **XMLTaskDataObjectDataType** complex type MUST NOT contain any elements or values.

The following table specifies the attributes of the **XMLTaskDataObjectDataType** complex type.

| Attribute | Description |
|---|---|
| OperationType | An enumeration value that specifies the type of operation that is performed by the XML Task. |
| SourceType | An enumeration value that specifies the source type for the XML document that is used by the XML Task. |
| Source | A string value that contains the source. The content of the string is dependent upon the value of the **SourceType** attribute. |
| SecondOperandType | An enumeration value that selects the source type of the second document that is used by the XML Task. |
| SecondOperand | A string value that contains the source for the second operation. The content of the string is dependent upon the value of the **SecondOperandType** attribute. |
| SaveOperationResult | An enumeration value that specifies whether to save the result of the operation. "True" specifies that the result of the operation is saved. "False" specifies that the result of the operation is not saved. |
| DestinationType | An enumeration value that specifies the type of the destination. |
| Destination | A string value that specifies the destination. The content of the string is dependent upon the value of the **DestinationType** attribute. |
| OverwriteDestination | An enumeration value that specifies whether to overwrite any content that already |

| Attribute | Description |
|---|---|
| | exists at the destination. |
| | "True" specifies that the existing content at the destination is overwritten. |
| | "False" specifies that existing content at the destination is not overwritten. |
| FailOpOnValidationFail | An enumeration value that specifies whether the operation fails if the document fails to validate. |
| | "True" specifies that the operation fails if the document fails to validate. |
| | "False" specifies that the operation does not necessarily fail if the document fails to validate. |
| FailOpOnDifference | An enumeration value that specifies whether to fail the operation if there is a difference detected. |
| | "True" specifies that the operation fails if there is a difference. |
| | "False" specifies that the operation does not necessarily fail if there is a difference. |
| SaveDiffGram | An enumeration value that specifies whether to save the result of a Diff operation as a DiffGram document. |
| | "True" specifies that the result can be saved as a DiffGram document. |
| | "False" specifies that the result is not saved. |
| DiffGramDestinationType | An enumeration value that specifies the type of the destination of the DiffGram output. |
| DiffGramDestination | A string value that specifies the destination of the DiffGram output. The content of the string varies with the value of the **DiffGramDestinationType** attribute value. |
| DiffOptions | An integer value that specifies the options to be used in the Diff operation. This integer is a bitmask value. |
| DiffAlgorithm | An enumeration value that specifies the algorithm to use for the Diff operation. |
| XPathStringSourceType | An enumeration value that specifies the type of the source of the XPATH string. |
| XPathStringSource | A string value that specifies the XPATH string source. The content of the string is dependent upon the value of the **XPathStringSourceType** attribute. |
| XPathOperation | An enumeration value that specifies the type of XPATH operation to perform. |
| PutResultInOneNode | An enumeration value that specifies whether to put the output result in one XML node. |
| | "True" specifies that the output result is put in one node. |
| | "False" specifies that the output result does not need to be put in one node. |
| ValidationType | An enumeration value that specifies the type of validation to perform. |

### 2.7.1.26.1    XMLTaskOperationTypeEnum

The **XMLTaskOperationTypeEnum** type contains the enumeration values for the **OperationType** attribute.

The following is the XSD of the **XMLTaskOperationTypeEnum** type.

```
<xs:simpleType name="XMLTaskOperationTypeEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Diff"/>
    <xs:enumeration value="Validate"/>
    <xs:enumeration value="Merge"/>
```

```
            <xs:enumeration value="XSLT"/>
            <xs:enumeration value="XPATH"/>
            <xs:enumeration value="Patch"/>
        </xs:restriction>
    </xs:simpleType>
```

The following table specifies the enumeration values for the **XMLTaskOperationTypeEnum** type.

| Enumeration value | Description |
|---|---|
| Diff | Specifies that two XML documents are compared. |
| Validate | Specifies that the XML is validated against an XSD or a DTD. |
| Merge | Specifies that two XML documents are merged. |
| XSLT | Specifies that an XSLT transformation is run on an XML document. |
| XPATH | Specifies that XPATH queries and evaluations are performed. |
| Patch | Specifies that the output of a Diff operation is applied to an XML document to create a new document. |

### 2.7.1.26.2   XMLTaskDiffOptionsType

The **XMLTaskDiffOptionsType** type is the type of the value that is used for the **DiffOptions** attribute. The **XMLTaskDiffOptionsType** type is a bitmask.

The following is the XSD of the **XMLTaskDiffOptionsType** type.

```
<xs:simpleType name="XMLTaskDiffOptionsType">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="255"/>
  </xs:restriction>
</xs:simpleType>
```

The bitmask value for the **XMLTaskDiffOptionsType** type is computed according to the following table. The table shows the value for each option that is enabled. The value for the **XMLTaskDiffOptionsType** type is the sum of the values for the options that are enabled. A value of "0" for the **XMLTaskDiffOptionsType** type means that none of the options in the table are selected.

| Value | Option |
|---|---|
| 1 | Specifies that the order of child nodes is ignored. |
| 2 | Specifies that comment nodes are not compared. |
| 4 | Specifies that processing instructions are not compared. |
| 8 | Specifies that white space is ignored. Significant white spaces are not compared, and all text nodes are normalized by discarding any leading and trailing white space characters (#x9, #x10, #x13, #x20) and by replacing sequences of white space characters with a single space (#x20) character. |
| 16 | Specifies that namespace names of elements and attributes are ignored. This option also implies that name prefixes are ignored. When this option is selected, two names that have the same local name but that have a different namespace URI and prefix are treated as the same names. |

| Value | Option |
|-------|--------|
| 32 | Specifies that the prefixes of element and attribute names are not compared. When this option is selected, two names that have the same local name and namespace URI but that have a different prefix are treated as the same names. |
| 64 | Specifies that the XML declaration is ignored. |
| 128 | Specifies that the Document Type Declaration (DTD) is ignored. |

### 2.7.1.26.3  XMLTaskDiffAlogrithmEnum

The **XMLTaskDiffAlgorithmEnum** type contains the enumeration values for the **DiffAlgorithm** attribute.

The following is the XSD of the **XMLTaskDiffAlgorithmEnum** type.

```
<xs:simpleType name="XMLTaskDiffAlgorithmEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Auto"/>
    <xs:enumeration value="Fast"/>
    <xs:enumeration value="Precise"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **XMLTaskDiffAlgorithmEnum** type.

| Enumeration value | Description |
|-------------------|-------------|
| Auto | Specifies that the XML Task chooses which algorithm to use. |
| Fast | Specifies that a fast, but less precise, algorithm is used. |
| Precise | Specifies that a precise diff algorithm is used. |

### 2.7.1.26.4  XMLTaskXPathOperationEnum

The **XMLTaskXPathOperationEnum** type contains the enumeration values for the **XPathOperation** attribute.

The following is the XSD of the **XMLTaskXPathOperationEnum** type.

```
<xs:simpleType name="XMLTaskXPathOperationEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Evaluation"/>
    <xs:enumeration value="NodeList"/>
    <xs:enumeration value="Values"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **XMLTaskXPathOperationEnum** type.

| Enumeration value | Description |
|---|---|
| Evaluation | Specifies that the results of an XPATH function are returned. |
| NodeList | Specifies that selected nodes of an XPATH operation are returned as an XML fragment. |
| Values | Specifies that the inner text value of all selected nodes are returned, concatenated into a string. |

### 2.7.1.26.5   XMLTaskValidationTypeEnum

The **XMLTaskValidationTypeEnum** type contains the enumeration values for the **ValidationType** attribute.

The following is the XSD of the **XMLTaskValidationTypeEnum** type.

```
<xs:simpleType name="XMLTaskValidationTypeEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="DTD"/>
    <xs:enumeration value="XSD"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **XMLTaskValidationTypeEnum** type.

| Enumeration value | Description |
|---|---|
| DTD | Specifies that a DTD is used to validate an XML document. |
| XSD | Specifies that an XSD is used to validate an XML document. |

## 2.8   Other Types for Package and Non-Package Executables

### 2.8.1   PropertyElementBaseType

A **Property** element is commonly used throughout the DTSX structure. The **PropertyElementBaseType** complex type provides a base type that many declarations of the **Property** element use for extension.

The following is the XSD of the **PropertyElementBaseType** complex type.

```
<xs:complexType name="PropertyElementBaseType">
  <xs:simpleContent>
    <xs:extension base="xs:anySimpleType">
      <xs:attribute name="DataType" type="DTS:DtsDataTypeEnum"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

Note that the XSD for the **PropertyElementBaseType** complex type allows the type of the value in the element to be **xs:anySimpleType**. However, in individual cases, there are constraints on what

the **Property** element value is allowed to be. The additional constraints MUST be followed. When such constraints are present, they are specified in the sections where the **Property** element is referenced.

The following table provides additional information about the attributes, types, and constraints for the **PropertyElementBaseType** complex type.

| Element | Constraints | Description |
|---------|-------------|-------------|
| DataType | If this optional attribute is specified, the data type of the **Property** element value MUST be of the specified **DataType**. Otherwise, the data type of the **Property** element value is **xs:anySimpleType**. | Specifies the data type of the **Property** element. |

## 2.8.1.1 DtsDataTypeEnum

The **DtsDataTypeEnum** type contains the values that can be specified for the **DTS:DataType** attribute. The value for the element for which a value from the **DtsDataTypeEnum** type is specified as the attribute value MUST be of the type that corresponds to the integer value of the enumeration as described in the following table.

The following is the XSD of the **DtsDataTypeEnum** type.

```
<xs:simpleType name="DtsDataTypeEnum">
  <xs:restriction base="xs:int">
    <xs:enumeration value="0"/>
    <xs:enumeration value="1"/>
    <xs:enumeration value="2"/>
    <xs:enumeration value="3"/>
    <xs:enumeration value="4"/>
    <xs:enumeration value="5"/>
    <xs:enumeration value="6"/>
    <xs:enumeration value="7"/>
    <xs:enumeration value="8"/>
    <xs:enumeration value="11"/>
    <xs:enumeration value="13"/>
    <xs:enumeration value="14"/>
    <xs:enumeration value="16"/>
    <xs:enumeration value="17"/>
    <xs:enumeration value="18"/>
    <xs:enumeration value="19"/>
    <xs:enumeration value="20"/>
    <xs:enumeration value="21"/>
    <xs:enumeration value="22"/>
    <xs:enumeration value="23"/>
    <xs:enumeration value="64"/>
    <xs:enumeration value="72"/>
    <xs:enumeration value="129"/>
    <xs:enumeration value="130"/>
    <xs:enumeration value="131"/>
    <xs:enumeration value="133"/>
    <xs:enumeration value="134"/>
    <xs:enumeration value="135"/>
    <xs:enumeration value="139"/>
    <xs:enumeration value="145"/>
    <xs:enumeration value="146"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **DtsDataTypeEnum** type.

| Enumeration value | Type for enumeration value |
|---|---|
| 0 | Empty value. |
| 1 | Null value. |
| 2 | Two-byte integer. |
| 3 | Four-byte integer. |
| 4 | Four-byte real. |
| 5 | Eight-byte real. |
| 6 | Currency. |
| 7 | Date. |
| 8 | BSTR. |
| 11 | Boolean. |
| 13 | Object. |
| 14 | Decimal. |
| 16 | One-byte integer. |
| 17 | One-byte unsigned integer. |
| 18 | Two-byte unsigned integer. |
| 19 | Four-byte unsigned integer. |
| 20 | Eight-byte integer. |
| 21 | Eight-byte unsigned integer. |
| 22 | Integer. |
| 23 | Unsigned integer. |
| 64 | An unsigned 64-bit date/time value. Represents the number of 100 nanosecond units since the start of January 1, 1601. |
| 72 | GUID. |
| 129 | A variable-length string with a specified maximum length. |
| 130 | Null-terminated Unicode character string with maximum length of 8000 characters. |
| 131 | Numeric. |
| 133 | **DbDate**, a structure that consists of year, month, day. |
| 134 | **DbTime**, a structure that consists of hour, minute, second. |
| 135 | **DbTimeStamp**, a structure that consists of year, month, day, hour, minute, seconds, and fractional parts. |
| 139 | **Varnumeric**, a structure that can hold very large numeric values. |
| 145 | Same as value 134, but includes fractional seconds. |

| Enumeration value | Type for enumeration value |
|---|---|
| 146 | Same as value 135, but includes a time zone offset. |

## 2.8.2 PropertyExpressionElementType

The **PropertyExpressionElementType** complex type specifies the type for the **PropertyExpression** element. The **PropertyExpression** element is commonly used throughout the DTSX structure.

The following is the XSD of the **PropertyExpressionElementType** complex type.

```
<xs:complexType name="PropertyExpressionElementType">
  <xs:simpleContent>
    <xs:extension base="xs:anySimpleType">
      <xs:attribute name="Name" type="xs:string" use="required"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

The following table provides additional information about the attributes and types for the **PropertyExpressionElementType** complex type.

| Attribute | Constraints | Description |
|---|---|---|
| Name | The string given MUST be one of the valid **Property** element **Name** attribute values for a **Property** element attribute on the same object. Or, the string given MUST be a property of a task executable that is defined within the **ObjectData** element of that task executable. | Specifies the **Name** attribute of the **PropertyExpression** element. Each named property has its own use and interpretation. |

## 2.8.3 LoggingOptionsType

The **LoggingOptionsType** complex type contains information about how logging and filtering of logging information is performed in the containing executable.

The following is the XSD of the **LoggingOptionsType** complex type.

```
<xs:complexType name="LoggingOptionsType">
  <xs:sequence>
    <xs:element name="Property" type="DTS:LoggingOptionsPropertyElementType"
                minOccurs="0" maxOccurs="unbounded">
    </xs:element>
    <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
                minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="SelectedLogProviders" minOccurs="0"
                type="DTS:SelectedLogProvidersType"/>
  </xs:sequence>
  <xs:attributeGroup ref="DTS:LoggingOptionsAttributeGroup"/>
</xs:complexType>
```

As the XSD shows, the **Property** element of the **LoggingOptionsType** complex type uses a different base type than the **Property** element does for most other objects in the DTSX structure.

The following table provides additional information about the elements, types, and constraints for the **LoggingOptionsType** complex type.

| Element | Additional constraints | Type definition | Description |
|---|---|---|---|
| Property | In some cases, constraints are placed on value of the **Property** element for specific values of the **Name** attribute of the **Property** element. If such constraints exist, they are specified where the meaning of the particular property name is specified. These constraints MUST be followed. If the optional **DataType** attribute is present, the value of the **Property** element MUST be of the type indicated by the **DataType** attribute. If the **Name** attribute of the **Property** is not equal to "ColumnFilter", the **Property** element MUST NOT have child elements. If the **Name** attribute equals "ColumnFilter", the **Property** element is a complex type. For more information, see Property Name=ColumnFilter. | Inline in the XSD in this section. | **Property** elements contain values for many different properties associated with an instance of the containing Executable element, according to the **Name** attribute of each **Property** element. |
| PropertyExpression | Additional constraints exist for the **PropertyExpression** element. These constraints MUST be followed. | PropertyExpressionElementType | The **PropertyExpression** element serves the same purpose as specified for the **Property** element except that an expression that is evaluated at run time is contained in the element's value that is stored in the Integration Services file format. |
| SelectedLogProviders | None | SelectedLogProvidersType | Contains a collection of the references to the log providers that are to be used. |

### 2.8.3.1 LoggingOptionsPropertyElementType

The **LoggingOptionsPropertyElementType** complex type is the type of the **Property** element when it appears as a child element to a **LoggingOptions** element. This type differs from the type for the **Property** element in most other places in this document.

The following is the XSD of the **LoggingOptionsPropertyElementType** complex type.

```
<xs:complexType name="LoggingOptionsPropertyElementType" mixed="true">
  <xs:sequence>
    <xs:element name="Property" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:anySimpleType">
            <xs:attribute name="Name" type="DTS:ColumnEventFilterPropertyNameEnum"
                          use="required"/>
            <xs:attribute name="DataType" type="DTS:DtsDataTypeEnum"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="Name" type="DTS:LoggingOptionsPropertyNameEnum"
                use="required"/>
  <xs:attribute name="DataType" type="DTS:DtsDataTypeEnum"/>
  <xs:attribute name="EventName" type="xs:string"/>
</xs:complexType>
```

The following table provides additional information about the attributes and types for the **Property** element and any **Property** element that is a child element to the **Property** element. In the case of **LoggingOptions**, the **Property** element is of type **LoggingOptionsPropertyElementType**, and these attributes are declared in that type definition.

| Attribute | Description |
|---|---|
| Name | Specifies the name of the **Property** element. Each named property has its own use and interpretation. The explanation for the interpretation of each named property is given with the definition of the type for the **Name** attribute. |
| DataType | The type of the value of the **Property** element. If present, the **Property** element value MUST be of this type. |
| EventName | The name of the event to which a column filter applies. |

When the **Property** element has a **Name** attribute value of "ColumnFilter", the **Property** element MUST be a complex type. This complex type contains additional **Property** elements. For more information, see Property Name=ColumnFilter.

### 2.8.3.1.1 LoggingOptionsPropertyNameEnum

The **LoggingOptionsPropertyNameEnum** type contains the values for the **Name** attribute of the **Property** element that are allowed for the **Property** element of the LoggingOptionsType complex type.

The following is the XSD of the **LoggingOptionsPropertyNameEnum** type.

```
<xs:simpleType name="LoggingOptionsPropertyNameEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="EventFilter"/>
    <xs:enumeration value="ColumnFilter"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **LoggingOptionsPropertyNameEnum** type.

| Property name | Description |
|---|---|
| EventFilter | A string that contains information about the events that are being filtered on. |
| ColumnFilter | Contains the definition of how the columns will be filtered. For additional restrictions, see Property Name=ColumnFilter. |

#### 2.8.3.1.1.1 Property Name=ColumnFilter

When the **Property** element **Name** attribute value is ColumnFilter, the **Property** element is a complex element and MUST contain additional child **Property** elements, as allowed for in the XSD. These child **Property** elements also have a **Name** attribute. The **Name** attribute for the child elements MUST be of type **ColumnEventFilterPropertyNameEnum**.

The following is the XSD of the **ColumnEventFilterPropertyNameEnum** type.

```
<xs:simpleType name="ColumnEventFilterPropertyNameEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Computer"/>
    <xs:enumeration value="Operator"/>
    <xs:enumeration value="SourceName"/>
    <xs:enumeration value="SourceID"/>
    <xs:enumeration value="ExecutionID"/>
    <xs:enumeration value="MessageText"/>
    <xs:enumeration value="DataBytes"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the interpretation of the values in the enumeration. For all of the **Property** Names listed in this table, the element value MUST be -1 for TRUE and 0 for FALSE.

| Property name | Description |
|---|---|
| Computer | Specifies whether the local computer name is included in the log entry. |
| Operator | Specifies whether the user account is included in the log entry. |
| SourceName | Specifies whether the name of the task that caused the event is included in the log entry. |
| SourceID | Specifies whether the GUID of the task that caused the event is included in the log entry. |
| ExecutionID | Specifies whether the unique package GUID is included in the log entry. |
| MessageText | Specifies whether message text is included in the log entry. |
| DataBytes | Specifies whether a byte array is included in the log entry. |

### 2.8.3.2 SelectedLogProvidersType

The **SelectedLogProvidersType** complex type is the container type for a collection of elements of type SelectedLogProviderType.

The following is the XSD of the **SelectedLogProvidersType** complex type.

```
<xs:complexType name="SelectedLogProvidersType">
  <xs:sequence>
    <xs:element name="SelectedLogProvider" type="DTS:SelectedLogProviderType"
                minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

The following table provides additional information about the elements, types, and constraints for the **SelectedLogProvidersType** complex type.

| Element | Type definition | Description |
|---|---|---|
| SelectedLogProvider | SelectedLogProviderType | Specifies a log provider that applies to the element that contains the collection. |

### 2.8.3.2.1 SelectedLogProviderType

The **SelectedLogProviderType** complex type is a reference to the log provider to be used for the logging that is selected within the **LoggingOptions** element.

The following is the XSD of the **SelectedLogProviderType** complex type.

```
<xs:complexType name="SelectedLogProviderType">
  <xs:sequence/>
  <xs:attribute name="InstanceID" type="DTS:uuid"/>
</xs:complexType>
```

The following table provides additional information about the attributes and types for the **SelectedLogProviderType** complex type.

| Attribute | Description |
|---|---|
| InstanceID | A reference to the **DTSID** property for a **LogProvider** that is contained in the package. The **DTSID** property is contained in the BasePropertyAttributeGroup attribute group. |

### 2.8.3.3 LoggingOptionsAttributeGroup

The **LoggingOptionsAttributeGroup** attribute group contains the attributes for the LoggingOptionsType complex type.

The following is the XSD for the **LoggingOptionsAttributeGroup** attribute group.

```
<xs:attributeGroup name="LoggingOptionsAttributeGroup">
  <xs:attribute name="LoggingMode" type="DTS:LoggingModeEnum" default="0"
                use="optional" form="qualified"/>
  <xs:attribute name="FilterKind" type="DTS:FilterKindEnum" default="1"
                use="optional" form="qualified"/>
  <xs:attribute name="EventFilter" type="xs:string" default=""
                use="optional" form="qualified"/>
</xs:attributeGroup>
```

The following table specifies the attributes for the **LoggingOptionsAttributeGroup** attribute group.

| Attribute | Description |
|-----------|-------------|
| LoggingMode | An enumeration value that specifies the logging mode. |
| FilterKind | An enumeration value that specifies whether filtering is done by inclusion or exclusion. |
| EventFilter | A string value that contains information about the events that are being filtered on. |

### 2.8.3.3.1 LoggingModeEnum

The **LoggingModeEnum** type contains the enumeration values that specify the available logging modes.

The following is the XSD of the **LoggingModeEnum** type.

```
<xs:simpleType name="LoggingModeEnum">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="2"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the descriptions for the enumeration values of the **LoggingModeEnum** type.

| Enumeration value | Type of logging mode specified |
|-------------------|-------------------------------|
| 0 | Use Parent Setting. |
| 1 | Enabled. |
| 2 | Disabled. |

### 2.8.3.3.2 FilterKindEnum

The **FilterKindEnum** type contains the enumeration values that specify whether filtering is done by inclusion or exclusion.

The following is the XSD of the **FilterKindEnum** type.

```
<xs:simpleType name="FilterKindEnum">
  <xs:restriction base="xs:int">
    <xs:enumeration value="0"/>
    <xs:enumeration value="1"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the descriptions for the enumeration values of the **FilterKindEnum** type.

| Enumeration value | Type of filtering specified |
|---|---|
| 0 | Filter by inclusion. |
| 1 | Filter by exclusion. |

### 2.8.4 VariablesType

The **VariablesType** complex type is the container type for a collection of elements of type VariableType.

The following is the XSD of the **VariablesType** type.

```
<xs:complexType name="VariablesType">
  <xs:sequence>
    <xs:element name="Variable" type="DTS:VariableType"
                minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

The following table provides additional information about the elements, types, and constraints for the **VariablesType** complex type.

| Element | Type definition | Description |
|---|---|---|
| Variable | VariableType | Specifies a variable that applies to the element that contains the collection. |

### 2.8.4.1 VariableType

The **VariableType** complex type is used to specify variables that are available to the package. The variables are evaluated at run time.

The following is the XSD of the **VariableType** complex type.

```
<xs:complexType name="VariableType">
  <xs:sequence>
    <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
                minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="VariableValue">
      <xs:complexType mixed="true">
        <xs:sequence>
          <xs:any minOccurs="0" maxOccurs="unbounded" processContents="skip"/>
        </xs:sequence>
        <xs:attribute name="DataType" type="DTS:DtsDataTypeEnum"/>
        <xs:attribute name="DataSubType">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="ManagedSerializable"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
  <xs:attributeGroup ref="DTS:VariableAttributeGroup"/>
```

```
            </xs:complexType>
```

The following table provides additional information about the elements, types, and constraints for the **VariableType** complex type.

| Element | Additional constraints | Type definition | Description |
|---|---|---|---|
| PropertyExpression | Additional constraints exist for the **PropertyExpression** element. These constraints MUST be followed. | PropertyExpressionElementType | The **PropertyExpression** element serves the same purpose as specified for the **Property** element, except that an expression that is evaluated at run time is contained in the element's value that is stored in the Integration Services file format. |
| VariableValue | If the optional **DataType** attribute is included, the value of the element MUST be of the type that is specified by the **DataType** attribute. If the optional **DataSubType** attribute is included, the **VariableValue** element can contain complex content. | | Contains the value that is used literally or is evaluated as an expression. |

The following table provides additional information about the attributes, attribute groups, and types for the **VariableType** type.

| Attribute | Description |
|---|---|
| BasePropertyAttributeGroup | An attribute group that specifies attributes that are common to many different types throughout the schema. |
| VariableAttributeGroup | An attribute group that specifies attributes for the **VariableType** type. |

The following table specifies the enumeration values for the **VariableType** type.

| Enumeration value | Description |
|---|---|
| ManagedSerializable | Specifies a managed and serializable object that can be serialized as XML in a DTSX file. This is the only valid value for the **DataSubType** attribute. |

### 2.8.4.1.1 VariableAttributeGroup

The **VariableAttributeGroup** attribute group contains the attributes for the VariableType type.

The following is the XSD for the **VariableAttributeGroup** attribute group.

```
<xs:attributeGroup name="VariableAttributeGroup">
  <xs:attribute name="Namespace" type="xs:string" use="optional"
              default="" form="qualified"/>
  <xs:attribute name="Expression" type="xs:string" use="optional"
```

```
                       default="" form="qualified"/>
        <xs:attribute name="EvaluateAsExpression" type="DTS:BooleanStringCap"
                       use="optional" default="False" form="qualified"/>
        <xs:attribute name="ReadOnly" type="DTS:BooleanStringCap" use="optional"
                       default="False" form="qualified"/>
        <xs:attribute name="RaiseChangedEvent" type="DTS:BooleanStringCap"
                       use="optional" default="False" form="qualified"/>
        <xs:attribute name="IncludeInDebugDump" type="xs:int" use="optional"
                       default="0" form="qualified"/>
    </xs:attributeGroup>
```

The following table specifies the attributes for the **VariableAttributeGroup** attribute group.

| Attribute | Description |
|---|---|
| Namespace | A string value that specifies the namespace that the variable belongs to. |
| Expression | A string value that specifies the expression for the variable. This value is used only if the **EvaluateAsExpression** property is set to value -1. |
| EvaluateAsExpression | A value of type **DTS:BooleanStringCap** that specifies whether the variable contains an expression. <br><br> "True" specifies that the variable contains an expression. <br><br> "False" specifies that the variable does not contain an expression. |
| ReadOnly | A value of type **DTS:BooleanStringCap** that specifies whether the variable is read-only. <br><br> "True" specifies that the variable is read-only. <br><br> "False" specifies that the variable is not read-only. |
| RaiseChangedEvent | A value of type **DTS:BooleanStringCap** that specifies whether the variable is flagged to raise the **OnVariableValueChanged** event when the variable's value changes. <br><br> "True" specifies that the variable is flagged to have the **OnVariableValueChanged** event raised when the variable's value changes. <br><br> "False" specifies that an **OnVariableValueChanged** event is not raised when the variable's value changes. |
| IncludeInDebugDump | An integer value that specifies whether the variable is included in a debug dump. |

#### 2.8.4.1.1.1   IncludeInDebugDumpEnum

The **IncludeInDebugDumpEnum** type contains the enumeration values that specify how whether the variable is included in a debug dump.

The following is the XSD of the **IncludeInDebugDumpEnum** type.

```
    <xs:simpleType name="IncludeInDebugDumpEnum">
      <xs:restriction base="xs:int">
        <xs:enumeration value="2345"/>
        <xs:enumeration value="6789"/>
      </xs:restriction>
    </xs:simpleType>
```

The following table specifies the descriptions for the enumeration values of the **IncludeInDebugDumpEnum** type.

| Enumeration value | Description |
|---|---|
| 2345 | Specifies TRUE; the variable is included in debug dumps. |
| 6789 | Specifies FALSE; the variable is not included in debug dumps. |

### 2.8.5 EventHandlersType

The **EventHandlersType** complex type is the container type for a collection of elements of type EventHandlerType.

The following is the XSD of the **EventHandlersType** complex type.

```
<xs:complexType name="EventHandlersType">
  <xs:sequence>
    <xs:element name="EventHandler" type="DTS:EventHandlerType"
                minOccurs="0" maxOccurs="unbounded" form="qualified"/>
  </xs:sequence>
</xs:complexType>
```

The following table provides additional information about the elements, types, and constraints for the **EventHandlersType** complex type.

| Element | Type definition | Description |
|---|---|---|
| EventHandler | EventHandlerType | Specifies an **EventHandler** element which in turn specifies the options, parameters, and settings for the event handler. |

### 2.8.5.1 EventHandlerType

The **EventHandlerType** complex type is used to specify the triggering events and the executables that are executed when a specific event occurs during the execution of a container, task, or package.

The following is the XSD of the **EventHandlerType** complex type.

```
<xs:complexType name="EventHandlerType">
  <xs:sequence>
    <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
                minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Variables" type="DTS:VariablesType" minOccurs="0"/>
    <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType"/>
    <xs:element name="Executables" type="DTS:ExecutablesType" minOccurs="0"/>
    <xs:element name="PrecedenceConstraints" type="DTS:PrecedenceConstraintsType"
                minOccurs="0"/>
  </xs:sequence>
  <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
  <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
  <xs:attributeGroup ref="DTS:EventHandlerAttributeGroup"/>
  <xs:attribute name="refId" type="xs:string" use="required" form="qualified"/>
</xs:complexType>
```

The following table provides additional information about the elements, types, and constraints for the **EventHandlerType** complex type.

| Element | Constraints | Type definition | Description |
|---------|-------------|-----------------|-------------|
| Variables | None | VariablesType | Specifies a collection of definitions of variables that can be defined in the context of the event handler. |
| LoggingOptions | None | LoggingOptionsType | Specifies information about what kind of information is logged for this event handler and where that information is recorded. |
| PropertyExpression | Additional constraints exist for the **PropertyExpression** element. | PropertyExpressionElementType | Serves the same purpose as specified for the **Property** element, except that an expression that is evaluated at run time is contained in the element's value that is stored in the Integration Services file format. |
| Executables | None | ExecutablesType | Specifies a collection of executables that are executed when the event handler is fired. |
| PrecedenceConstraints | None | PrecedenceConstraintsType | Specifies a collection of precedence constraints between contained executables in this **EventHandler** element. |

The following table provides additional information about the attributes, attribute groups, and types for the **EventHandlerType** complex type.

| Attribute | Description |
|-----------|-------------|
| BasePropertyAttributeGroup | An attribute group that specifies attributes that are common to many different types throughout the schema. |
| BaseExecutablePropertyAttributeGroup | An attribute group that specifies attributes that are common to many different types throughout the schema. |
| EventHandlerAttributeGroup | An attribute group that specifies attributes for the **EventHandlerType** complex type. |
| refId | A string value that specifies a unique **refId** for this object. Elsewhere within the containing executable or within the package, the value of **refId** can be used to refer to this object.<104> |

### 2.8.5.1.1 EventHandlerAttributeGroup

The **EventHandlerAttributeGroup** attribute group contains the attributes for the EventHandlerType type.

The following is the XSD for the **EventHandlerAttributeGroup** attribute group.

```
<xs:attributeGroup name="EventHandlerAttributeGroup">
  <xs:attribute name="EventID" type="xs:int" use="optional"
                default="0" form="qualified"/>
  <xs:attribute name="EventName" type="DTS:EventNameEnum"
                use="required" form="qualified"/>
</xs:attributeGroup>
```

The following table specifies the attributes for the **EventHandlerAttributeGroup** attribute group.

| Attribute | Description |
|-----------|-------------|
| EventID | Specifies an integer that identifies this event.<105> |
| EventName | An enumeration value that specifies the name of the event that triggers this event handler. |

### 2.8.5.1.1.1 EventNameEnum

The **EventNameEnum** type contains the enumeration values that specify the name of the event that will trigger the event handler.

The following is the XSD of the **EventNameEnum** type.

```
<xs:simpleType name="EventNameEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="OnError"/>
    <xs:enumeration value="OnExecStatusChanged"/>
    <xs:enumeration value="OnInformation"/>
    <xs:enumeration value="OnPostExecute"/>
    <xs:enumeration value="OnPreExecute"/>
    <xs:enumeration value="OnPreValidate"/>
    <xs:enumeration value="OnProgress"/>
    <xs:enumeration value="OnQueryCancel"/>
    <xs:enumeration value="OnTaskFailed"/>
    <xs:enumeration value="OnVariableValueChanged"/>
    <xs:enumeration value="OnWarning"/>
    <xs:enumeration value="OnCustomEvent"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the descriptions for the enumeration values of the **EventNameEnum** type.

| Enumeration value | Description |
|-------------------|-------------|
| OnError | Specifies that the event is fired by an error condition. |
| OnExecStatusChanged | Specifies that the event is fired if execution status changes. |
| OnInformation | Specifies that the event is fired by availability of information. |
| OnPostExecute | Specifies that the event is fired after execution is complete. |
| OnPreExecute | Specifies that the event is fired before execution begins. |
| OnPreValidate | Specifies that the event is fired before validation begins. |
| OnProgress | Specifies that the event is fired when a progress status update is available. |

| Enumeration value | Description |
|---|---|
| OnQueryCancel | Specifies that the event is fired when a query is canceled. |
| OnTaskFailed | Specifies that the event is fired when a task fails. |
| OnVariableValueChanged | Specifies that the event is fired when a variable value changes. |
| OnWarning | Specifies that the event is fired if a warning condition is encountered. |
| OnCustomEvent | Specifies that the event is fired as defined by a custom event. |

## 2.8.6 PrecedenceConstraintsType

The **PrecedenceConstraintsType** complex type is the container type for a collection of elements of type PrecedenceConstraintType.

The following is the XSD of the **PrecedenceConstraintsType** complex type.

```
<xs:complexType name="PrecedenceConstraintsType">
  <xs:sequence>
    <xs:element name="PrecedenceConstraint" type="DTS:PrecedenceConstraintType"
            minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

The following table provides additional information about the elements, types, and constraints for the **PrecedenceConstraintsType** complex type.

| Element | Type definition | Description |
|---|---|---|
| PrecedenceConstraint | PrecedenceConstraintType | Specifies a precedence constraint that applies to the element that contains the collection. |

### 2.8.6.1 PrecedenceConstraintType

The **PrecedenceConstraintType** complex type is used to specify an object that contains precedence constraints between the executable elements within a package.

The following is the XSD of the **PrecedenceConstraintType** complex type.

```
<xs:complexType name="PrecedenceConstraintType">
  <xs:sequence>
    <xs:element name="PropertyExpression" type="DTS:PropertyExpressionElementType"
            minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
  <xs:attributeGroup ref="DTS:PrecedenceConstraintAttributeGroup"/>
  <xs:attribute name="refId" type="xs:string" use="required" form="qualified"/>
</xs:complexType>
```

The following table provides additional information about the elements, types, and constraints for the **PrecedenceConstraintType** complex type.

| Element | Additional constraints | Type definition | Description |
|---------|------------------------|-----------------|-------------|
| PropertyExpression | Additional constraints exist for the **PropertyExpression** element. | PropertyExpressionElementType | The **PropertyExpression** element serves the same purpose as specified for the **Property** element except that an expression that is evaluated at run time is contained in the element's value that is stored in the Integration Services file format. |

The following table provides additional information about the attributes, attribute groups, and types for the **PrecedenceConstraintType** complex type.

| Attribute | Description |
|-----------|-------------|
| BasePropertyAttributeGroup | An attribute group that specifies attributes that are common to many different types throughout the schema. |
| PrecedenceConstraintAttributeGroup | An attribute group that specifies attributes for the **PrecedenceConstraintType** complex type. |
| refId | A string value that specifies a unique **refId** for this object. Elsewhere within the containing executable or within the package, the value of **refId** can be used to refer to this object.<106> |

#### 2.8.6.1.1 PrecedenceConstraintAttributeGroup

The **PrecedenceConstraintAttributeGroup** attribute group contains the attributes for the PrecedenceConstraintType type.

The following is the XSD for the **PrecedenceConstraintAttributeGroup** attribute group.

```
<xs:attributeGroup name="PrecedenceConstraintAttributeGroup">
  <xs:attribute name="Value" type="DTS:PrecedenceConstraintValueEnum"
                use="optional" default="0" form="qualified"/>
  <xs:attribute name="EvalOp" type="DTS:PrecedenceConstraintEvalOpEnum"
                use="optional" default="1" form="qualified"/>
  <xs:attribute name="LogicalAnd" type="DTS:BooleanStringCap"
                use="optional" default="False" form="qualified"/>
  <xs:attribute name="Expression" type="xs:string" use="optional"
                default="" form="qualified"/>
  <xs:attribute name="From" type="xs:string" use="required"
                form="qualified"/>
  <xs:attribute name="To" type="xs:string" use="required"
                form="qualified"/>
</xs:attributeGroup>
```

The following table specifies the attributes for the **PrecedenceConstraintAttributeGroup** attribute group.

| Attribute | Description |
|-----------|-------------|
| Value | An enumeration value that specifies the task completion status that is being checked for to determine precedence. |

| Attribute | Description |
|-----------|-------------|
| EvalOp | An enumeration value that specifies the type of evaluation operation for this **PrecedenceConstraint** instance. |
| LogicalAnd | A value of type **DTS:BooleanStringCap** that specifies whether to perform a logical **AND** operation on all the precedence constraints for an executable.<br><br>"True" specifies that the precedence operation is a logical **AND** that includes all the precedence constraints for an executable.<br><br>"False" specifies that the precedence operation is not a logical **AND** between all the precedence constraints for an executable. |
| Expression | A string value that specifies an expression to be evaluated. |
| From | A string value that specifies a **refId** attribute value on an Executable element in the package that is the origin end of the precedence constraint. |
| To | A string value that specifies a **refId** attribute value on an **Executable** element in the package that is the destination end of the precedence constraint. |

#### 2.8.6.1.1.1    PrecedenceConstraintEvalOpEnum

The **PrecedenceConstraintEvalOpEnum** type contains the enumeration values that specify the type of evaluation operation for the precedence constraint.

The following is the XSD of the **PrecedenceConstraintEvalOpEnum** type.

```
<xs:simpleType name="PrecedenceConstraintEvalOpEnum">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="1"/>
    <xs:maxInclusive value="4"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the descriptions for the enumeration values of the **PrecedenceConstraintEvalOpEnum** type.

| Enumeration value | Description |
|-------------------|-------------|
| 1 | Specifies that the evaluation operation is based on an expression. |
| 2 | Specifies that the evaluation operation is based on a constraint. |
| 3 | Specifies that the evaluation operation is based on an expression and a constraint. |
| 4 | Specifies that the evaluation operation is based on an expression or a constraint. |

#### 2.8.6.1.1.2    PrecedenceConstraintValueEnum

The **PrecedenceConstraintValueEnum** type contains the enumeration values that specify the type of evaluation operation for the precedence constraint.

The following is the XSD of the **PrecedenceConstraintValueEnum** type.

```
<xs:simpleType name="PrecedenceConstraintValueEnum">
```

```
        <xs:restriction base="xs:int">
          <xs:minInclusive value="0"/>
          <xs:maxInclusive value="2"/>
        </xs:restriction>
     </xs:simpleType>
```

The following table specifies the descriptions for the enumeration values of the
**PrecedenceConstraintValueEnum** type.

| Enumeration value | Description |
|---|---|
| 0 | The task completed with a return value of Success. |
| 1 | The task completed with a return value of Failure. |
| 2 | The task ran to completion. |

## 2.9 Enumeration Types and Supporting Types

### 2.9.1 BasePropertyAttributeGroup

The **BasePropertyAttributeGroup** attribute group contains attributes that are used by many objects
in the structure.

The following is the XSD for the **BasePropertyAttributeGroup** attribute group.

```
<xs:attributeGroup name="BasePropertyAttributeGroup">
  <xs:attribute name="Description" type="xs:string" default=""
                use="optional" form="qualified"/>
  <xs:attribute name="DTSID" type="DTS:uuid" use="required" form="qualified"/>
  <xs:attribute name="CreationName" type="xs:string" use="required" form="qualified"/>
  <xs:attribute name="ObjectName" type="xs:string" use="optional" form="qualified"/>
</xs:attributeGroup>
```

The following table specifies the attributes for the **BasePropertyAttributeGroup** attribute group.

| Attribute | Description |
|---|---|
| Description | A string value that specifies a human-readable descriptive string.<107> |
| DTSID | A value of type **DTS:uuid** that specifies a unique identifier for this object. Elsewhere within the containing executable or within the package, the value of **DTSID** can be used to refer to this object.<108> |
| CreationName | A string value that specifies the **CreationName** for this object.<109> |
| ObjectName | A string value that specifies the **ObjectName** for this object.<110> |

### 2.9.2 BaseExecutablePropertyAttributeGroup

The **BaseExecutablePropertyAttributeGroup** attribute group contains attributes that are used by
executable objects in the structure.

The following is the XSD for the **BaseExecutablePropertyAttributeGroup** attribute group.

```
<xs:attributeGroup name="BaseExecutablePropertyAttributeGroup">
  <xs:attribute name="ForceExecValue" type="DTS:BooleanStringCap"
                default="False" use="optional" form="qualified"/>
  <xs:attribute name="ExecValue" type="xs:anySimpleType" default="0"
                use="optional" form="qualified"/>
  <xs:attribute name="ForceExecutionResult" default="-1" use="optional"
                type="DTS:ForceExecutionResultEnum" form="qualified"/>
  <xs:attribute name="Disabled" type="DTS:BooleanStringCap" default="False"
                use="optional" form="qualified"/>
  <xs:attribute name="FailPackageOnFailure" type="DTS:BooleanStringCap"
                default="False" use="optional" form="qualified"/>
  <xs:attribute name="FailParentOnFailure" type="DTS:BooleanStringCap"
                default="False" use="optional" form="qualified"/>
  <xs:attribute name="MaxErrorCount" type="xs:int" default="1"
                use="optional" form="qualified"/>
  <xs:attribute name="ISOLevel" type="DTS:ISOLevelEnum"
                default="1048576" use="optional" form="qualified"/>
  <xs:attribute name="LocaleID" type="xs:int" use="optional" form="qualified"/>
  <xs:attribute name="TransactionOption" type="DTS:TransactionOptionEnum"
                default="1" use="optional" form="qualified"/>
  <xs:attribute name="DelayValidation" type="DTS:BooleanStringCap"
                default="False" use="optional" form="qualified"/>
  <xs:attribute name="ExecValueType" type="DTS:DtsDataTypeEnum"
                use="required" form="qualified"/>
</xs:attributeGroup>
```

The following table specifies the attributes for the **BaseExecutablePropertyAttributeGroup** attribute group.

| Attribute | Description |
|---|---|
| ForceExecValue | A value of type **DTS:BooleanStringCap** that specifies whether the Executable is forced to return a value regardless of the exit status of the executable. <br><br>"True" specifies that the **Executable** is forced to return a value regardless of exit status. <br><br>"False" specifies that the **Executable** is not forced to return a value regardless of the exit status. |
| ExecValue | A value of type **xs:anySimpleType** that specifies the value that is returned if the value of the **ForceExecValue** attribute is "True". |
| ForceExecutionResult | An enumeration value that specifies the return result that is returned if a result is forced to be returned, instead of the actual result. |
| Disabled | A value of type **DTS:BooleanStringCap** that specifies whether the executable is disabled and not available at run time. <br><br>"True" specifies that the executable is disabled and not available at run time. <br><br>"False" specifies that the executable is enabled. |
| FailPackageOnFailure | A value of type **DTS:BooleanStringCap** that specifies whether the package fails if this executable fails. <br><br>"True" specifies that the package fails if the executable fails. <br><br>"False" specifies that the package does not necessarily fail if the executable fails. |
| FailParentOnFailure | A value of type **DTS:BooleanStringCap** that specifies whether the parent of the containing **Executable** element fails if this executable fails. <br><br>"True" specifies that the parent executable fails if this executable fails. <br><br>"False" specifies that the parent executable does not necessarily fail if this executable fails. |
| MaxErrorCount | An integer value that specifies the maximum number of errors before the executable fails. If this count is reached or exceeded, the executable MUST fail. |

| Attribute | Description |
|---|---|
| ISOLevel | An enumeration value that specifies the isolation level for the executable. |
| LocaleID | An integer value that specifies the LCID for the language for this executable.<111> |
| TransactionOption | An enumeration value that specifies the transaction support and the transaction enrollment of the container. |
| DelayValidation | A value of type **DTS:BooleanStringCap** that specifies whether the validation of the executable is delayed until its container executes.<br>"True" specifies that the validation of the executable is delayed until run time.<br>"False" specifies that the validation of the executable is not delayed until run time. |
| ExecValueType | A value of type **DTS:DtsDataTypeEnum** that specifies the type of **ExecValue** if the value of the **ForceExecValue** attribute is "True". |

### 2.9.2.1  ISOLevelEnum

The **ISOLevelEnum** type contains the enumeration values that specify isolation levels that can be specified.

The following is the XSD of the **ISOLevelEnum** type.

```
<xs:simpleType name="ISOLevelEnum">
  <xs:restriction base="xs:int">
    <xs:enumeration value="-1"/>
    <xs:enumeration value="16"/>
    <xs:enumeration value="1033"/>
    <xs:enumeration value="4096"/>
    <xs:enumeration value="65536"/>
    <xs:enumeration value="1048576"/>
    <xs:enumeration value="16777216"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the descriptions for the enumeration values of the **ISOLevelEnum** type.

| Enumeration value | Type of isolation level |
|---|---|
| 1048576 | Serializable |
| 16777216 | Snapshot |
| 65536 | Repeatable Read |
| 1033 | Read Uncommitted |
| 4096 | Read Committed |
| 16 | Chaos |
| -1 | Unspecified |

### 2.9.2.2 TransactionOptionEnum

The **TransactionOptionEnum** type contains the enumeration values that specify transaction support options.

The following is the XSD of the **TransactionOptionEnum** type.

```
<xs:simpleType name="TransactionOptionEnum">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="2"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the descriptions for the enumeration values of the **TransactionOptionEnum** type.

| Enumeration value | Support for transactions indicated |
|---|---|
| 0 | Not supported |
| 1 | Supported |
| 2 | Required |

### 2.9.2.3 ForceExecutionResultEnum

The **ForceExecutionResultEnum** type contains the enumeration values that specify the result values that can be specified to be returned, instead of the actual result value, when a value's return is forced.

The following is the XSD of the **ForceExecutionResultEnum** type.

```
<xs:simpleType name="ForceExecutionResultEnum">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="-1"/>
    <xs:maxInclusive value="2"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the descriptions for the enumeration values of the **ForceExecutionResultEnum** type.

| Enumeration value | Execution result that is forced |
|---|---|
| -1 | None |
| 0 | Success |
| 1 | Failure |
| 2 | Completion |

### 2.9.3 AllExecutableAttributeGroup

The **AllExecutableAttributeGroup** attribute group contains attributes that are used by all executable objects in the structure.

The following is the XSD for the **AllExecutableAttributeGroup** attribute group.

```
<xs:attributeGroup name="AllExecutableAttributeGroup">
  <xs:attribute name="DisableEventHandlers" type="DTS:BooleanStringCap"
                default="False" use="optional" form="qualified"/>
</xs:attributeGroup>
```

The following table specifies the attributes for the **AllExecutableAttributeGroup** attribute group.

| Attribute | Description |
|---|---|
| DisableEventHandlers | A value of type **DTS:BooleanStringCap** that specifies whether event handlers for this executable are disabled. <br><br>"True" specifies that any event handlers applied to this executable are disabled. <br><br>"False" specifies that event handlers applied to this executable are not disabled. |

### 2.9.4 BooleanStringCap

The **BooleanStringCap** type contains string enumeration values that are used to act as a Boolean value and to express a true or false state. The **BooleanStringCap** type is referenced by other types in the Integration Services package file format.

The following is the XSD for the **BooleanStringCap** type.

```
<xs:simpleType name="BooleanStringCap">
  <xs:restriction base="xs:string">
    <xs:enumeration value="True"/>
    <xs:enumeration value="False"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **BooleanStringCap** type.

| Enumeration value | Description |
|---|---|
| False | Specifies that the setting is False. |
| True | Specifies that the setting is True. |

### 2.9.5 uuid

The **uuid** type is used to specify the text pattern for references to DTSID values.

The following is the XSD of the **uuid** type.

```
<xs:simpleType name="uuid">
  <xs:restriction base="xs:string">
```

```
            <xs:pattern value="\{[0-9a-zA-Z]{8}-[0-9a-zA-Z]{4}-[0-9a-zA-Z]{4}-[0-9a-zA-Z]{4}-
                [0-9a-zA-Z]{12}\}" />
        </xs:restriction>
    </xs:simpleType>
```

### 2.9.6 PasswordElementType

The **PasswordElementType** complex type specifies the type of the **PassWord** element that is a child element of other elements within the Integration Services package file format. The **PasswordElementType** type is used to store the password information for connecting to a data source.

The following is the XSD of the **PasswordElementType** complex type.

```
<xs:complexType name="PasswordElementType">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="Sensitive" form="unqualified">
        <xs:simpleType>
          <xs:restriction base="xs:int">
            <xs:minInclusive value="0"/>
            <xs:maxInclusive value="1"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="Encrypted" form="unqualified">
        <xs:simpleType>
          <xs:restriction base="xs:int">
            <xs:minInclusive value="0"/>
            <xs:maxInclusive value="1"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

The **PasswordElementType** complex type MUST NOT contain any elements.

The following table specifies the attributes for the **PasswordElementType** complex type.

| Attribute | Description |
|---|---|
| Sensitive | Specifies that the element value contains an indicator for whether the password information is treated as sensitive. The integer value has the following interpretations: 0 = FALSE; the password is not considered to be sensitive. 1 = TRUE; the password is considered to be sensitive. |
| Encrypted | Specifies that the element value contains an indicator for whether the element value is encrypted. The integer value has the following interpretations: 0 = FALSE; the element value is not encrypted. 1 = TRUE; the element value is encrypted. |

### 2.9.7 SourceTypeEnum

The **SourceTypeEnum** type contains enumeration values that are used to specify the type of an input source. The **SourceTypeEnum** type is referenced by other types in the Integration Services package file format.

The following is the XSD for the **SourceTypeEnum** type.

```
<xs:simpleType name="SourceTypeEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="DirectInput"/>
    <xs:enumeration value="FileConnection"/>
    <xs:enumeration value="Variable"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **SourceTypeEnum** type.

| Enumeration value | Description |
|---|---|
| DirectInput | Specifies that the source text is contained inline in a package file as the value of another element. |
| FileConnection | Specifies that the source text is contained in a file connection. |
| Variable | Specifies that the source text is contained in a variable. |

### 2.9.8 DestinationTypeEnum

The **DestinationTypeEnum** type contains enumeration values that are used to specify the type of a destination source. The **DestinationTypeEnum** type is referenced by other types in the Integration Services package file format.

The following is the XSD of the **DestinationTypeEnum** type.

```
<xs:simpleType name="DestinationTypeEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="FileConnection"/>
    <xs:enumeration value="Variable"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **DestinationTypeEnum** type.

| Enumeration value | Description |
|---|---|
| FileConnection | Specifies that the source is contained in a file connection. |
| Variable | Specifies that the source is contained in a variable. |

### 2.9.9 IfObjectExistsEnum

The **IfObjectExistsEnum** type contains the enumeration values for the **IfObjectExists** attribute.

The following is the XSD of the **IfObjectExistsEnum** type.

```
<xs:simpleType name="IfObjectExistsEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="FailTask"/>
    <xs:enumeration value="Overwrite"/>
    <xs:enumeration value="Skip"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the enumeration values for the **IfObjectExistsEnum** type.

| Enumeration value | Description |
|---|---|
| FailTask | Specifies that the task MUST fail if the object exists at the destination for the transfer. |
| Overwrite | Specifies that the task MUST overwrite the previously existing object if the object exists at the destination for the transfer. |
| Skip | Specifies that the task MUST skip the transfer of such objects if the object exists at the destination for the transfer. |

## 2.9.10 InnerObjectObjectDataType

The **InnerObjectObjectDataType** complex type specifies a type that can contain any element. It is used for extending and customizing Executable elements and **LogProvider** elements.

The following is the XSD of the **InnerObjectObjectDataType** complex type.

```
<xs:complexType name="InnerObjectObjectDataType">
  <xs:sequence>
    <xs:any minOccurs="0" maxOccurs="unbounded"
            namespace="##any" processContents="skip"/>
  </xs:sequence>
</xs:complexType>
```

The **InnerObjectObjectDataType** type MAY contain any element in any namespace.<112>

## 2.9.11 DesignTimeProperties String Value XML Content

The **DesignTimeProperties** element is of type **xs:string**. The contents of the string value MUST form a valid XML document with an **Objects** root node. This section specifies the XML for the **Objects** node, specified as XSD definitons. However, none of these elements appear as XML elements within a package file, as they appear only within the string value for the **DesignTimeProperties** element.

Where elements within the **DesignTimeProperties** string value refer to elements in a namespace, this is documented. All other elements are in an unspecified namespace.

### 2.9.11.1    Objects Element

The declaration for the **Objects** element, the root node of the DesignTimeProperties string value, is as follows.

```
<xs:element name="Objects" type="ObjectsType"/>
```

#### 2.9.11.1.1 ObjectsType

The **ObjectsType** complex type specifies the screen layout of a package diagram.

The following is the XSD for the **ObjectsType** complex type.

```
<xs:complexType name="ObjectsType">
  <xs:sequence>
    <xs:element name="Package" type="DTS:DesignTimePropertiesContainerType"/>
    <xs:element name="TaskHost" type="DTS:DesignTimePropertiesContainerType"
                minOccurs="0"/>
    <xs:element name="DtsEventHandler" type="DTS:DesignTimePropertiesContainerType"
                minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="Version" type="xs:string"/>
</xs:complexType>
```

The following table provides additional information about the elements for the **ObjectsType** complex type.

| Element | Description |
|---|---|
| Package | A complex type that specifies the screen layout of a package. |
| TaskHost | A complex type that specifies the screen layout of the tasks within a package. |
| DtsEventHandler | A complex type that specifies the screen layout of event handlers within a package. |

The following table provides additional information about the attributes for the **ObjectsType** complex type.

| Attribute | Description |
|---|---|
| Version | A string value that specifies information about the product version that the described objects correspond to. |

##### 2.9.11.1.1.1 DesignTimePropertiesContainerType

The **DesignTimePropertiesContainerType** complex type specifies the screen layout of one of several individual object types used to diagram a package.

The following is the XSD for the **DesignTimePropertiesContainerType** complex type.

```
<xs:complexType name="DesignTimePropertiesContainerType">
  <xs:sequence>
    <xs:element name="LayoutInfo" type="LayoutInfoType"/>
  </xs:sequence>
  <xs:attribute name="design-time-name" type="xs:string"/>
</xs:complexType>
```

The following table provides additional information about the elements for the **DesignTimePropertiesContainerType** complex type.

| Element | Description |
|---------|-------------|
| LayoutInfo | A complex type that specifies screen layout information for each object. |

The following table provides additional information about the attributes for the
**DesignTimePropertiesContainerType** complex type.

| Attribute | Description |
|-----------|-------------|
| design-time-name | A string value that specifies a symbolic name that is used to identify the object. |

### 2.9.11.1.1.1.1.1    LayoutInfoType

The **LayoutInfoType** complex type specifies the screen layout of an individual object from a session
of using the SSIS Designer user interface.

The following is the XSD for the **LayoutInfoType** complex type.

```
<xs:complexType name="LayoutInfoType">
  <xs:sequence>
    <xs:element  ref="is-ser:GraphLayout" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

The following table provides additional information about the elements for the **LayoutInfoType**
complex type.

| Element | Description |
|---------|-------------|
| is-ser:GraphLayout | A complex type that specifies the screen layout of an object. |

The **LayoutInfoType** complex type MUST NOT have attributes.

### 2.9.11.1.1.1.1.1.1    is-ser Namespace

The namespace that is identified by the **is-ser** prefix is defined as follows.

```
xmlns:is-ser=
"clr-namespace:Microsoft.SqlServer.IntegrationServices.Designer.
Model.Serialization;assembly=Microsoft.SqlServer.
IntegrationServices.Graph"
```

### 2.9.11.1.1.1.1.1.2    is-ser:GraphLayout Element

The **is-ser:GraphLayout** element is defined as follows.

```
<xs:element name="GraphLayout" type="is-ser:GraphLayoutType"/>
```

### 2.9.11.1.1.1.1.1.3    GraphLayoutType

The **GraphLayoutType** complex type specifies the layout graph of the objects that describe a
package.

The following is the XSD for the **GraphLayoutType** complex type.

```
<xs:complexType name="GraphLayoutType">
  <xs:sequence>
    <xs:element name="NodeLayout" type="is-ser:NodeLayoutType"
                maxOccurs="unbounded" minOccurs="0"/>
    <xs:element name="EdgeLayout" type="is-ser:EdgeLayoutType"
                maxOccurs="unbounded" minOccurs="0"/>
    <xs:element name="ContainerLayout" type="is-ser:ContainerLayoutType"
                maxOccurs="unbounded" minOccurs="0"/>
    <xs:element name="AnnotationLayout" type="is-ser:AnnotationLayoutType"
                maxOccurs="unbounded" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="Capacity" type="xs:int"/>
</xs:complexType>
```

The following table provides additional information about the elements for the **GraphLayoutType** complex type.

| Element | Description |
|---|---|
| NodeLayout | A complex type that specifies the screen layout of a graph node. |
| EdgeLayout | A complex type that specifies the screen layout of an edge. |
| ContainerLayout | A complex type that specifies the screen layout of a container. |
| AnnotationLayout | A complex type that specifies the screen layout of an annotation. |

The following table provides additional information about the attributes for the **GraphLayoutType** complex type.

| Attribute | Description |
|---|---|
| Capacity | An integer value that specifies the maximum number of contained elements that can be held without resizing the structure. |

### 2.9.11.1.1.1.1.3.1 NodeLayoutType

The **NodeLayoutType** complex type specifies the screen layout of an individual node of a package.

The following is the XSD for the **NodeLayoutType** complex type.

```
<xs:complexType name="NodeLayoutType">
  <xs:sequence/>
  <xs:attribute name="Size" type="xs:string"/>
  <xs:attribute name="Id" type="xs:string"/>
  <xs:attribute name="TopLeft" type="xs:string"/>
</xs:complexType>
```

The **NodeLayoutType** complex type contains no elements.

The following table provides additional information about the attributes for the **NodeLayoutType** complex type.

| Attribute | Description |
|-----------|-------------|
| Size | A string value that specifies the size of the node, in Windows Presentation Foundation (WPF) units. |
| Id | A string value that specifies an identifier for the node. |
| TopLeft | A string value that specifies the coordinates of the top-left corner of the node, in WPF units. |

#### 2.9.11.1.1.1.1.3.2 EdgeLayoutType

The **EdgeLayoutType** complex type specifies the screen layout of an edge within a package layout diagram.

The following is the XSD for the **EdgeLayoutType** complex type.

```
<xs:complexType name="EdgeLayoutType">
  <xs:sequence>
    <xs:element name="EdgeLayout.Curve" type="is-ser:EdgeLayoutCurveType" />
    <xs:element name="EdgeLayout.Labels" type="is-ser:EdgeLayoutLabelsType" />
  </xs:sequence>
  <xs:attribute name="Id" type="xs:string"/>
  <xs:attribute name="TopLeft" type="xs:string"/>
</xs:complexType>
```

The following table provides additional information about the elements for the **EdgeLayoutType** complex type.

| Element | Description |
|---------|-------------|
| EdgeLayout.Curve | A complex type that describes the layout of the curve associated with an edge. |
| EdgeLayout.Labels | A complex type that describes the layout of the labels associated with an edge. |

The following table provides additional information about the attributes for the **EdgeLayoutType** complex type.

| Attribute | Description |
|-----------|-------------|
| Id | A string value that specifies an identifier for the edge. |
| TopLeft | A string value that specifies the top-left coordinate of the edge, in WPF units. |

#### 2.9.11.1.1.1.1.3.3 EdgeLayoutCurveType

The **EdgeLayoutCurveType** complex type specifies the screen layout of a curve that is part of a package representation.

The following is the XSD for the **EdgeLayoutCurveType** complex type.

```
<xs:complexType name="EdgeLayoutCurveType">
  <xs:sequence>
    <xs:element ref="mssgle:Curve"/>
  </xs:sequence>
  <xs:attribute name="Id" type="xs:string"/>
  <xs:attribute name="TopLeft" type="xs:string"/>
```

```
                  </xs:complexType>
```

The following table provides additional information about the elements for the **EdgeLayoutCurveType** complex type.

| Element | Description |
|---------|-------------|
| mssgle:Curve | A complex type defined in the namespace referenced by the **mssgle** prefix that specifies the segments of the curve. |

The following table provides additional information about the attributes for the **EdgeLayoutCurveType** complex type.

| Attribute | Description |
|-----------|-------------|
| Id | A string value that specifies an identifier for the curve. |
| TopLeft | A string value that specifies the top-left coordinate of the curve, in WPF units. |

#### 2.9.11.1.1.1.1.3.4   mssgle Namespace

The namespace identified by the **mssgle** prefix is defined as follows.

```
xmlns:mssgle="clr-namespace:Microsoft.SqlServer.Graph.LayoutEngine;
assembly=Microsoft.SqlServer.Graph"
```

#### 2.9.11.1.1.1.1.3.5   mssgle:Curve Element

The **mssgle:Curve** element is defined as follows.

```
<xs:element name="Curve" type="mssgle:CurveType"/>
```

#### 2.9.11.1.1.1.1.3.6   CurveType

The **CurveType** complex type specifies the screen layout of a curve that is part of a package representation.

The following is the XSD for the **CurveType** complex type.

```
<xs:complexType name="CurveType">
  <xs:sequence>
    <xs:element name="Curve.Segments" type="mssgle:CurveSegmentsType" />
  </xs:sequence>
  <xs:attribute name="StartConnector" type="xs:string"/>
  <xs:attribute name="EndConnector" type="xs:string"/>
  <xs:attribute name="Start" type="xs:string"/>
  <xs:attribute name="End" type="xs:string"/>
</xs:complexType>
```

The following table provides additional information about the elements for the **CurveType** complex type.

| Element | Description |
|---|---|
| Curve.Segments | A complex type defined in the namespace referenced by the **mssgle** prefix that specifies an individual segment of the curve. |

The following table provides additional information about the attributes for the **CurveType** complex type.

| Attribute | Description |
|---|---|
| StartConnector | A string value that specifies the coordinates of the start point of the start connector for the curve, in WPF units. |
| EndConnector | A string value that specifies the coordinates of the end point of the end connector for the curve, in WPF units. |
| Start | A string value that specifies the coordinates of the start point of the curve, in WPF units. |
| End | A string value that specifies the coordinates of the end point of the curve, in WPF units. |

### 2.9.11.1.1.1.1.3.7   CurveSegmentsType

The **CurveSegmentsType** complex type specifies a collection of segments of a curve as laid out on the screen.

The following is the XSD for the **CurveSegmentsType** complex type.

```
<xs:complexType name="CurveSegmentsType">
  <xs:sequence>
    <xs:element name="SegmentCollection" type="mssgle:SegmentCollectionType"/>
  </xs:sequence>
</xs:complexType>
```

The following table provides additional information about the elements for the **CurveSegmentsType** complex type.

| Element | Description |
|---|---|
| SegmentCollection | A complex type that specifies the collection of segments that form a curve. |

The **CurveSegmentsType** complex type has no attributes.

### 2.9.11.1.1.1.1.3.8   SegmentCollectionType

The **SegmentCollectionType** complex type specifies an individual segment of a curve.

The following is the XSD for the **SegmentCollectionType** complex type.

```
<xs:complexType name="SegmentCollectionType">
  <xs:sequence>
    <xs:element name="LineSegment" type="mssgle:LineSegmentType"
                minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="CubicBezierSegment" type="mssgle:CubicBezierSegmentType"
                minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="Capacity" type="xs:int"/>
```

```
        </xs:complexType>
```

The following table provides additional information about the elements for the
**SegmentCollectionType** complex type.

| Element | Description |
|---|---|
| LineSegment | A complex type that specifies a line segment. |
| CubicBezierSegment | A complex type that specifies a cubic Bezier curve segment. |

The following table provides additional information about the attributes for the
**SegmentCollectionType** complex type.

| Attribute | Description |
|---|---|
| Capacity | An integer value that specifies the maximum number of contained elements that can be held without resizing the structure. |

### 2.9.11.1.1.1.1.3.9   LineSegmentType

The **LineSegmentType** complex type specifies an individual line segment of a curve.

The following is the XSD for the **LineSegmentType** complex type.

```
<xs:complexType name="LineSegmentType">
  <xs:sequence/>
  <xs:attribute name="End" type="xs:string"/>
</xs:complexType>
```

The **LineSegmentType** complex type contains no elements.

The following table provides additional information about the attributes for the **LineSegmentType**
complex type.

| Attribute | Description |
|---|---|
| End | A string value that specifies the coordinates, in WPF units, of the end of the line segment. |

### 2.9.11.1.1.1.1.3.10  CubicBezierSegmentType

The **CubicBezierSegmentType** complex type specifies a cubic Bezier segment of a curve.

The following is the XSD for the **CubicBezierSegmentType** complex type.

```
<xs:complexType name="CubicBezierSegmentType">
  <xs:sequence/>
  <xs:attribute name="Point1" type="xs:string"/>
  <xs:attribute name="Point2" type="xs:string"/>
  <xs:attribute name="Point3" type="xs:string"/>
</xs:complexType>
```

The **CubicBezierSegmentType** complex type contains no elements.

The following table provides additional information about the attributes for the **CubicBezierSegmentType** complex type.

| Attribute | Description |
|---|---|
| Point1 | A string value that specifies the coordinates, in WPF units, of the first point of the cubic Bezier cube segment. |
| Point2 | A string value that specifies the coordinates, in WPF units, of the second point of the cubic Bezier cube segment. |
| Point3 | A string value that specifies the coordinates, in WPF units, of the third point of the cubic Bezier cube segment. |

### 2.9.11.1.1.1.1.3.11 EdgeLayoutLabelsType

The **EdgeLayoutLabelsType** complex type specifies a collection of edge layout labels of a curve as laid out on the screen.

The following is the XSD for the **EdgeLayoutLabelsType** complex type.

```
<xs:complexType name="EdgeLayoutLabelsType">
  <xs:sequence>
    <xs:element name="EdgeLabelCollection" type="is-ser:EdgeLabelCollectionType"
                minOccurs="0"/>
    <xs:element ref="mssgm:EdgeLabel" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

The following table provides additional information about the elements for the **EdgeLayoutLabelsType** complex type.

| Element | Description |
|---|---|
| EdgeLabelCollection | A complex type that specifies a collection of edge labels. |
| EdgeLabel | A complex type that specifies an individual edge label. |

The **EdgeLayoutLabelsType** complex type has no attributes.

### 2.9.11.1.1.1.1.3.12 EdgeLabelCollectionType

The **EdgeLabelCollectionType** complex type specifies a collection of a package as viewed on the screen.

The following is the XSD for the **EdgeLabelCollectionType** complex type.

```
<xs:complexType name="EdgeLabelCollectionType">
  <xs:sequence/>
</xs:complexType>
```

The **EdgeLabelCollectionType** complex type has no elements.

The **EdgeLabelCollectionType** complex type has no attributes.

### 2.9.11.1.1.1.1.3.13 mssgm Namespace

The namespace identified by the **mssgm** prefix is defined as follows.

```
xmlns:mssgm="clr-namespace:Microsoft.SqlServer.Graph.Model;
assembly=Microsoft.SqlServer.Graph"
```

### 2.9.11.1.1.1.1.3.14 mssgm:EdgeLabel Element

The **mssgm:EdgeLabel** element is defined as follows.

```
<xs:element name="EdgeLabel" type="mssgm:EdgeLabelType"/>
```

### 2.9.11.1.1.1.1.3.15 EdgeLabelType

The **EdgeLabelType** complex type specifies an edge label that is part of the screen representation of a package.

The following is the XSD for the **EdgeLabelType** complex type.

```
<xs:complexType name="EdgeLabelType">
  <xs:sequence/>
  <xs:attribute name="BoundingBox" type="xs:string"/>
  <xs:attribute name="RelativePosition" type="mssgm:RelativePositionEnum"/>
</xs:complexType>
```

The **EdgeLabelType** complex type contains no elements.

The following table provides additional information about the attributes for the **EdgeLabelType** complex type.

| Attribute | Description |
|---|---|
| BoundingBox | A string value that specifies the coordinates of the four vertices of the bounding box for the edge label, in WPF units. |
| RelativePosition | An enumeration value that specifies the relative position of the edge label. |

### 2.9.11.1.1.1.1.3.16 RelativePositionEnum

The **RelativePositionEnum** type contains the enumeration values that specify the allowed relative position values.

The following is the XSD of the **RelativePositionEnum** type.

```
<xs:simpleType name="RelativePositionEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Any"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the descriptions for the enumeration values of the **RelativePositionEnum** type.

| Enumeration value | Description |
|---|---|
| Any | The configuration is retrieved from any relative position. |

### 2.9.11.1.1.1.1.3.17 ContainerLayoutType

The **ContainerLayoutType** complex type specifies the screen layout of a container within a package description.

The following is the XSD for the **ContainerLayoutType** complex type.

```
<xs:complexType name="ContainerLayoutType">
  <xs:sequence/>
  <xs:attribute name="Size" type="xs:string"/>
  <xs:attribute name="Id" type="xs:string"/>
  <xs:attribute name="TopLeft" type="xs:string"/>
  <xs:attribute name="HeaderHeight" type="xs:double"/>
  <xs:attribute name="IsExpanded" type="DTS:BooleanStringCap"/>
  <xs:attribute name="PanelSize" type="xs:string"/>
</xs:complexType>
```

The **ContainerLayoutType** complex type contains no elements.

The following table provides additional information about the attributes for the **ContainerLayoutType** complex type.

| Attribute | Description |
|---|---|
| Size | A string value that specifies the size of the container, in WPF units. |
| Id | A string value that specifies an identifier for the container. |
| TopLeft | A string value that specifies the coordinates of the top-left corner of the container, in WPF units. |
| HeaderHeight | A value of type **xs:double** that specifies the height of the container header, in WPF units. |
| IsExpanded | A value of type **DTS:BooleanStringCap** that specifies whether the container is expanded or collapsed.<br>"True" specifies that the container is expanded.<br>"False" specifies that the container is collapsed. |
| PanelSize | A string value that specifies the panel size, in WPF units. |

### 2.9.11.1.1.1.1.3.18 AnnotationLayoutType

The **AnnotationLayoutType** complex type specifies the screen layout of an annotation that can be added to a package file layout.

The following is the XSD for the **AnnotationLayoutType** complex type.

```
<xs:complexType name="AnnotationLayoutType">
  <xs:sequence>
    <xs:element name="AnnotationLayout.FontInfo"
                type="is-ser:AnnotationLayoutFontInfoType"
                minOccurs="0"/>
  </xs:sequence>
```

```
            <xs:attribute name="ParentId" type="xs:string"/>
            <xs:attribute name="Text" type="xs:string"/>
            <xs:attribute name="Size" type="xs:string"/>
            <xs:attribute name="TopLeft" type="xs:string"/>
            <xs:attribute name="Id" type="xs:string"/>
        </xs:complexType>
```

The following table provides additional information about the elements for the **AnnotationLayoutType** complex type.

| Element | Description |
|---------|-------------|
| AnnotationLayout.FontInfo | A complex type that specifies information about the fonts used on the screen for an annotation. |

The following table provides additional information about the attributes for the **AnnotationLayoutType** complex type.

| Attribute | Description |
|-----------|-------------|
| ParentId | A string value that specifies the **Id** attribute value for the parent of this annotation. |
| Text | A string value that specifies the text of the annotation. |
| Size | A string value that specifies the size of the annotation, in WPF units. |
| TopLeft | A string value that specifies the top left coordinates of the annotation, in WPF units. |
| Id | A string value that specifies an identifier for this annotation. |

### 2.9.11.1.1.1.1.3.19 AnnotationLayoutFontInfoType

The **AnnotationLayoutFontInfoType** complex type specifies the fonts for an annotation to a package file.

The following is the XSD for the **AnnotationLayoutFontInfoType** complex type.

```
    <xs:complexType name="AnnotationLayoutFontInfoType">
      <xs:sequence>
        <xs:element ref="mssge:FontInfo"/>
      </xs:sequence>
    </xs:complexType>
```

The following table provides additional information about the elements for the **AnnotationLayoutFontInfoType** complex type.

| Element | Description |
|---------|-------------|
| FontInfo | A complex type that specifies information about the fonts used on the screen for an annotation. The **FontInfo** element is in the namespace designated by the prefix **mssge**. |

The **AnnotationLayoutFontInfoType** complex type has no attributes.

### 2.9.11.1.1.1.1.3.20 mssge Namespace

The namespace identified by the **mssge** prefix is defined as follows.

```
xmlns:mssge="clr-namespace:Microsoft.SqlServer.Graph.Extended;
assembly=Microsoft.SqlServer.IntegrationServices.Graph"
```

### 2.9.11.1.1.1.1.3.21 mssge:FontInfo Element

The **mssge:FontInfo** element is defined as follows.

```
<xs:element name="FontInfo" type="mssge:FontInfo"/>
```

### 2.9.11.1.1.1.1.3.22 FontInfoType

The **FontInfoType** complex type specifies the information about fonts as they appear on the screen.

The following is the XSD for the **FontInfoType** complex type.

```
<xs:complexType name="FontInfoType">
  <xs:sequence>
    <xs:element name="FontInfo.TextDecorations"
                type="mssge:FontInfoTextDecorationsType"/>
  </xs:sequence>
  <xs:attribute name="Family" type="xs:string"/>
  <xs:attribute name="Size" type="xs:int"/>
  <xs:attribute name="Color" type="xs:string"/>
  <xs:attribute name="Style" type="xs:string"/>
  <xs:attribute name="Weight" type="xs:string"/>
</xs:complexType>
```

The following table provides additional information about the elements for the **FontInfoType** complex type.

| Element | Description |
|---|---|
| FontInfo.TextDecorations | A complex type that specifies information about the text decorations added to a font. |

The following table provides additional information about the attributes for the **FontInfoType** complex type.

| Attribute | Description |
|---|---|
| Family | A string value that specifies the name of the font family. |
| Size | An integer value that specifies the size of the font. |
| Color | A string value that specifies the color of the font. The string value is the color value in hexadecimal; the prefix of this color value is the number sign (#) character. |
| Style | A string value that specifies the style for the font. |
| Weight | A string value that specifies the weight of the font. |

### 2.9.11.1.1.1.1.3.23 FontInfoTextDecorationsType

The **FontInfoTextDecorationsType** complex type specifies information about the text decorations that are added to a font.

The following is the XSD for the **FontInfoTextDecorationsType** complex type.

```
<xs:complexType name="FontInfoTextDecorationsType">
  <xs:sequence>
    <xs:element ref="av:TextDecorationCollection"/>
  </xs:sequence>
</xs:complexType>
```

The following table provides additional information about the elements for the **FontInfoTextDecorationsType** complex type.

| Element | Description |
|---|---|
| TextDecorationsCollection | A complex type that specifies information about the decorations that are added to the fonts used on the screen. The **TextDecorationsCollection** element is in the namespace that is designated by the prefix **av**. |

The **FontInfoTextDecorationsType** complex type has no attributes.

### 2.9.11.1.1.1.1.3.24 av Namespace

The namespace identified by the **av** prefix is defined as follows.

```
xmlns:av="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
```

### 2.9.11.1.1.1.1.3.25 av:TextDecorationsCollection Element

The **av:TextDecorationsCollection** element is defined as follows.

```
<xs:element name="TextDecorationCollection" type="av:TextDecorationCollectionType"/>
```

### 2.9.11.1.1.1.1.3.26 TextDecorationCollectionType

The **TextDecorationCollectionType** complex type specifies a text decoration for text that is part of the representation of a package file.

The following is the XSD for the **TextDecorationCollectionType** complex type.

```
<xs:complexType name="TextDecorationCollectionType">
  <xs:sequence>
    <xs:element name="TextDecoration" type="av:TextDecorationType"
                minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

The following table provides additional information about the elements for the **TextDecorationCollectionType** complex type.

| Element | Description |
|---|---|
| TextDecoration | A complex type that specifies information for a specific text decoration for a font. |

The **TextDecorationCollectionType** complex type has no attributes.

### 2.9.11.1.1.1.1.3.27 TextDecorationType

The **TextDecorationType** complex type specifies information for a specific text decoration for a font.

The following is the XSD for the **TextDecorationType** complex type.

```
<xs:complexType name="TextDecorationType">
  <xs:sequence/>
  <xs:attribute name="Location" type="xs:string"/>
</xs:complexType>
```

The **TextDecorationType** complex type has no elements.

The following table provides additional information about the attributes for the **TextDecorationType** complex type.

| Attribute | Description |
|-----------|-------------|
| Location | A string value that specifies the location of the text decoration. |

## 2.10 Custom Packages

Integration Services allows users to create custom packages that execute new objects beyond the objects that ship with the standard Integration Services system. These packages are described by the XSD in section 2 of this document. Users can create the following custom objects:

- Executable

- ConnectionManager

- Log provider

- ForEachEnumerator

- Component

The following sections describe these custom objects and their impact on the XSD of the DTSX structure format.

**Note**  For more information about the environment for the development of custom package objects, see [MSDN-EPCO].

### 2.10.1 Custom Executable

Software developers can author a custom executable by using a software development environment and a programming language. These custom executables can then be executed as part of a Integration Services package. When a custom executable is authored, the XSD for the Integration Services format structure can change in the following way:

- The **ObjectData** element of the executable can contain any XML that the executable author defines to support the authored executable.

The following section explains the impact on the XSD for custom executables.

#### 2.10.1.1      ExecutableObjectDataType Type Replacement

In the case of custom packages, the definition of the ExecutableObjectDataType complex type is replaced with the following XSD definition for the **ExecutableObjectDataTypeCustom** complex type.

All references to the **ExecutableObjectDataType** complex type are replaced with a reference to the **ExecutableObjectDataTypeCustom** complex type. The original **ExecutableObjectDataType** complex type has an XSD **choice** element, which allows the choice among elements specific to the variety of executable types that ship with the Integration Services system. In the **ExecutableObjectDataTypeCustom** complex type, the choice is constrained to one specific element from within the original choice, the **InnerObject** element. The Integration Services system does not interpret the contents of the **InnerObject** element, and it passes the contents to the custom executable when asked to do so.

The following is the XSD for the **ExecutableObjectDataTypeCustom** complex type.

```
<xs:complexType name="ExecutableObjectDataTypeCustom">
  <xs:choice>
    <xs:element name="InnerObject" type="DTS:InnerObjectObjectDataType"
                form="unqualified"/>
  </xs:choice>
</xs:complexType>
```

## 2.10.2 Custom ConnectionManager

Software developers can use a software development environment and a programming language to author a custom ConnectionManager to connect to a data source that is not supported by the Integration Services product. The custom connection manager is referenced from within the Integration Services package file like any other connection manager. When it is necessary to connect to the external data source, Integration Services executes the custom-authored code to do so.

To execute the custom connection manager code, the **DTSID** attribute of the **ConnectionManager** MUST be set to the GUID that references the registered COM component that implements the required interfaces or MUST be set to the full path of the Microsoft .NET Framework assembly.

The following sections explain the impact on the XSD for custom executables.

### 2.10.2.1 ConnectionManagerObjectDataType Type Replacement

In the case of a custom ConnectionManager, the definition of the ConnectionManagerObjectDataType complex type is replaced with the following XSD definition for the **ConnectionManagerObjectDataTypeCustom** complex type. All references to the **ConnectionManagerObjectDataType** complex type are replaced with a reference to the **ConnectionManagerObjectDataTypeCustom** complex type. The Integration Services system does not interpret the contents of the **ObjectData** element, and it passes the contents to the custom connection manager when asked to do so.

The following is the XSD for the **ConnectionManagerObjectDataTypeCustom** complex type.

```
<xs:complexType name="ConnectionManagerObjectDataTypeCustom">
  <xs:sequence>
    <xs:any minOccurs="0" maxOccurs="unbounded" namespace="##any"
        processContents="skip"/>
  </xs:sequence>
</xs:complexType>
```

## 2.10.3 Custom Log Provider

Software developers can use a software development environment and a programming language to author a custom log provider to connect to and use a logging service that is not supported by the Integration Services product. The custom log provider is referenced from within the Integration

Services package file like any other log provider. When it is necessary to access the log provider, Integration Services executes the custom-authored code to do so.

To execute the custom log provider code, the **DTSID** attribute of the **LogProvider** element MUST be set to the GUID that references the registered COM component that implements the required interfaces or MUST be set to the full path of the the .NET Framework assembly.

Custom log providers impose no changes on the XSD in this document because the LogProviderObjectDataType type already accommodates the presence of a custom log provider.

### 2.10.4 Custom ForEachEnumerator

Software developers can use a software development environment and a programming language to author a custom **ForEachEnumerator** to enumerate over objects for which a standard enumerator is not provided by the Integration Services product.

To execute the custom **ForEachEnumerator** code, the **DTSID** attribute of the **ForEachEnumerator** MUST be set to the GUID that references the registered COM component that implements the required interfaces or MUST be set to the full path of the .NET Framework assembly.

The following section explains the impact on the XSD for the custom ForEachEnumerator.

#### 2.10.4.1 ForEachEnumeratorObjectDataType Type Replacement

In the case of a custom **ForEachEnumerator**, the definition of the ForEachEnumeratorObjectDataType complex type is replaced with the following XSD definition for the **ForEachEnumeratorObjectDataTypeCustom** complextype. All references to the **ForEachEnumeratorObjectDataType** complex type are replaced with a reference to the **ForEachEnumeratorObjectDataTypeCustom** complex type. The Integration Services system does not interpret the contents of the **ObjectData** element, and it passes the contents to the custom connection manager when asked to do so.

The following is the XSD for the **ForEachEnumeratorObjectDataTypeCustom** complex type.

```
<xs:complexType name="ForEachEnumeratorObjectDataTypeCustom">
  <xs:sequence>
    <xs:any minOccurs="0" maxOccurs="unbounded"
            namespace="##any" processContents="skip"/>
  </xs:sequence>
</xs:complexType>
```

### 2.10.5 Custom Component

Software developers can use a software development environment and a programming language to author a custom component that will run within a Pipeline Task executable. When a custom component is authored, the author can define additional properties that are persisted in the XML beyond the properties that ship with Integration Services. The following sections define the impact on the XSD to accommodate development of custom components.

#### 2.10.5.1 PipelineComponentComponentClassIDEnum Type Replacement

To execute the custom component, the value of the **ComponentClassID** attribute of the component element MUST point to the custom code executable. The type of the **ComponentClassID** attribute MUST be changed to be the **PipelineComponentComponentClassIDEnumCustom** type. All references within the XSD to the PipelineComponentComponentClassIDEnum type MUST be changed to reference the **PipelineComponentComponentClassIDEnumCustom** type.

The following is the XSD for the **PipelineComponentComponentClassIDEnumCustom** type**.**

```
<xs:simpleType name="PipelineComponentComponentClassIDEnumCustom">
  <xs:union memberTypes="DTS:PipelineComponentComponentClassIDEnum">
    <xs:simpleType>
      <xs:restriction base="xs:string"/>
    </xs:simpleType>
  </xs:union>
</xs:simpleType>
```

### 2.10.5.2 PipelineComponentPropertyNameEnum Type Replacement

In a custom component, the developer can add additional properties that appear in the properties collection that is a child collection of the **component** element. To accommodate the presence of these additional properties, the PipelineComponentPropertyNameEnum type is replaced by the following definition of the **PipelineComponentPropertyNameEnumCustom** type. All references within the XSD to the **PipelineComponentPropertyNameEnum** type MUST be changed to reference the **PipelineComponentPropertyNameEnumCustom** type to accommodate the presence of custom components in the package file.

The following is the XSD for the **PipelineComponentPropertyNameEnumCustom** type.

```
<xs:simpleType name="PipelineComponentPropertyNameEnumCustom">
  <xs:union memberTypes="DTS:PipelineComponentPropertyNameEnum">
    <xs:simpleType>
      <xs:restriction base="xs:string"/>
    </xs:simpleType>
  </xs:union>
</xs:simpleType>
```

### 2.10.5.3 PipelineComponentInputPropertyNameEnum Type Replacement

In a custom component, the developer can add additional properties that appear in the properties collection that is a child collection of the **input** element. To accommodate the presence of these additional properties, the PipelineComponentInputPropertyNameEnum type is replaced by the following definition of the **PipelineComponentInputPropertyNameEnumCustom** type. All references within the XSD to the **PipelineComponentInputPropertyNameEnum** type MUST be changed to reference the **PipelineComponentInputPropertyNameEnumCustom** type to accommodate the presence of custom components in the package file.

The following is the XSD for the **PipelineComponentInputPropertyNameEnumCustom** type.

```
<xs:simpleType name="PipelineComponentInputPropertyNameEnumCustom">
  <xs:union memberTypes="DTS:PipelineComponentInputPropertyNameEnum">
    <xs:simpleType>
      <xs:restriction base="xs:string"/>
    </xs:simpleType>
  </xs:union>
</xs:simpleType>
```

### 2.10.5.4 PipelineComponentOutputPropertyNameEnum Type Replacement

In a custom component, the developer can add additional properties that appear in the properties collection that is a child collection of the **output** element. To accommodate the presence of these additional properties, the PipelineComponentOutputPropertyNameEnum type is replaced by the following definition of the **PipelineComponentOutputPropertyNameEnumCustom** type. All references within the XSD to the **PipelineComponentOutputPropertyNameEnum** type MUST be

changed to reference the **PipelineComponentOutputPropertyNameEnumCustom** type to accommodate the presence of custom components in the package file.

The following is the XSD for the **PipelineComponentOutputPropertyNameEnumCustom** type.

```
<xs:simpleType name="PipelineComponentOutputPropertyNameEnumCustom">
  <xs:union memberTypes="DTS:PipelineComponentOutputPropertyNameEnum">
    <xs:simpleType>
      <xs:restriction base="xs:string"/>
    </xs:simpleType>
  </xs:union>
</xs:simpleType>
```

### 2.10.5.5 PipelineComponentInputColumnPropertyNameEnum Type Replacement

In a custom component, the developer can add additional properties that appear in the properties collection that is a child collection of the **inputColumn** element. To accommodate the presence of these additional properties, the **PipelineComponentInputColumnPropertyNameEnum** type is replaced by the following definition of the **PipelineComponentInputColumnPropertyNameEnumCustom** type. All references within the XSD to the **PipelineComponentInputColumnPropertyNameEnum** type MUST be changed to reference the **PipelineComponentInputColumnPropertyNameEnumCustom** type to accommodate the presence of custom components in the package file.

The following is the XSD for the **PipelineComponentInputColumnPropertyNameEnumCustom** type.

```
<xs:simpleType
  name="PipelineComponentInputColumnPropertyNameEnumCustom">
  <xs:union memberTypes="DTS:PipelineComponentInputColumnPropertyNameEnum">
    <xs:simpleType>
      <xs:restriction base="xs:string"/>
    </xs:simpleType>
  </xs:union>
</xs:simpleType>
```

### 2.10.5.6 PipelineComponentOutputColumnPropertyNameEnum Type Replacement

In a custom component, the developer can add additional properties that appear in the properties collection that is a child collection of the **outputColumn** element. To accommodate the presence of these additional properties, the **PipelineComponentOutputColumnPropertyNameEnum** type is replaced by the following definition of the **PipelineComponentOutputColumnPropertyNameEnumCustom** type. All references within the XSD to the **PipelineComponentOutputColumnPropertyNameEnum** type MUST be changed to reference the **PipelineComponentOutputColumnPropertyNameEnumCustom** type to accommodate the presence of custom components in the package file.

The following is the XSD for the **PipelineComponentOutputColumnPropertyNameEnumCustom** type.

```
<xs:simpleType
  name="PipelineComponentOutputColumnPropertyNameEnumCustom">
  <xs:union memberTypes="DTS:PipelineComponentOutputColumnPropertyNameEnum">
    <xs:simpleType>
      <xs:restriction base="xs:string"/>
    </xs:simpleType>
  </xs:union>
</xs:simpleType>
```

# 3   Structure Examples

## 3.1   Lookup and Fuzzy Lookup Components with Error Output

The following example is taken from [Haselden], with modifications.

The following package file illustrates a Pipeline Task that contains several components within the pipeline. The Microsoft SQL Server Integration Services (SSIS) user interface in the designer that represents this package example is shown in the following diagram.
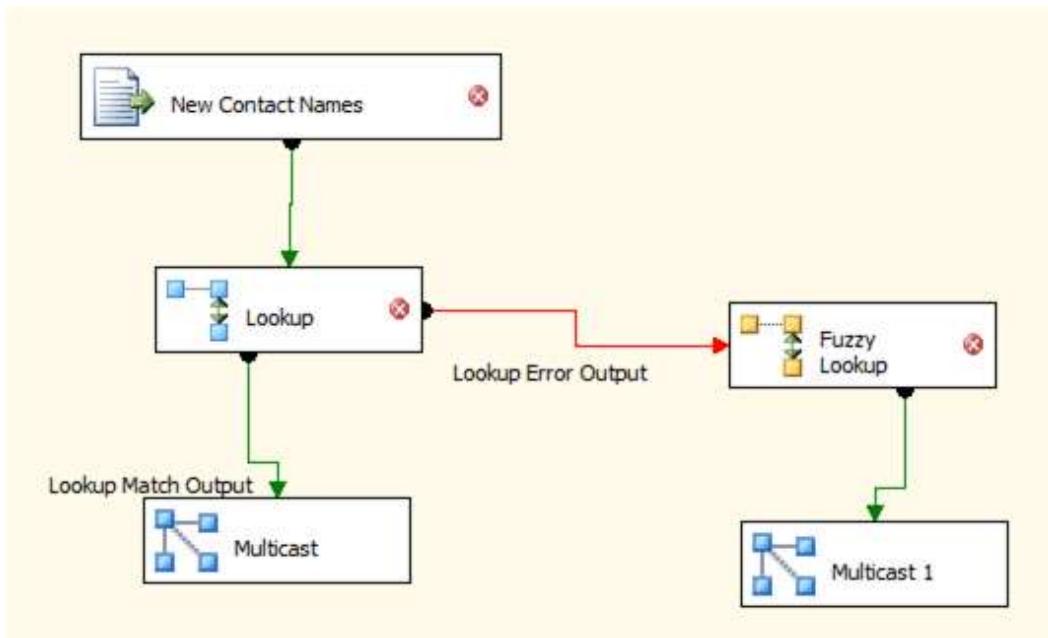


**Figure 3: Pipeline Task in Integration Services user interface**

The Lookup Component has both a standard output and an error output. In the case of error output from the Lookup Component, the data flow path is different from the case in which the lookup operation succeeds. In the following structure example, the **inputs** element and the **outputs** element of the Lookup Component illustrate its two different output paths.

```
<?xml version="1.0"?>
<DTS:Executable xmlns:DTS="www.microsoft.com/SqlServer/Dts"
  DTS:refId="Package"
  DTS:CreationDate="7/16/2005 3:59:24 PM"
  DTS:CreationName="SSIS.Package.3"
  DTS:CreatorComputerName="ROMEO"
  DTS:CreatorName="Kirk Haselden"
  DTS:DTSID="{ECF155AC-F433-4930-A0A1-BE9B065E004D}"
  DTS:EnableConfig="True"
  DTS:ExecutableType="SSIS.Package.3"
  DTS:ExecValue="0"
  DTS:ExecValueType="3"
  DTS:LastModifiedProductVersion="11.0.1153.1"
  DTS:LocaleID="1033"
  DTS:ObjectName="FuzzyLookup"
  DTS:PackageType="5"
  DTS:ProtectionLevel="0"
  DTS:VersionBuild="51"
  DTS:VersionGUID="{D4BC51B6-573E-48DA-B5E3-9F08625687A6}">
  <DTS:Property
    DTS:Name="PackageFormatVersion">6</DTS:Property>
```

```
<DTS:ConnectionManagers>
  <DTS:ConnectionManager
    DTS:refId="Package.ConnectionManagers[.\SQL2K5.AdventureWorks]"
    DTS:CreationName="OLEDB"
    DTS:DTSID="{FAFAC1E8-81D4-4E0D-A5A9-ABAD9BF38826}"
    DTS:ObjectName=".\SQL2K5.AdventureWorks">
    <DTS:PropertyExpression
      DTS:Name="ServerName">@[SAMPLES::SERVERNAME]</DTS:PropertyExpression>
    <DTS:ObjectData>
      <DTS:ConnectionManager
        DTS:ConnectionString="Data Source=.;Initial Catalog=AdventureWorks;
        Provider=SQLNCLI11;Integrated Security=SSPI;Auto Translate=False;" />
    </DTS:ObjectData>
  </DTS:ConnectionManager>
  <DTS:ConnectionManager
    DTS:refId="Package.ConnectionManagers[New Contacts]"
    DTS:CreationName="FLATFILE"
    DTS:DTSID="{E97F2191-17D8-400B-9EE6-7823482A958C}"
    DTS:ObjectName="New Contacts">
    <DTS:PropertyExpression
      DTS:Name="ConnectionString">@[SAMPLES::ROOTWD] +
      "\\SAMPLEDATA\\NewContacts.txt"</DTS:PropertyExpression>
    <DTS:ObjectData>
      <DTS:ConnectionManager
        DTS:Format="Delimited"
        DTS:LocaleID="1033"
        DTS:HeaderRowDelimiter="_x000D__x000A_"
        DTS:ColumnNamesInFirstDataRow="True"
        DTS:RowDelimiter=""
        DTS:TextQualifier="_x003C_none_x003E_"
        DTS:CodePage="1252"
        DTS:ConnectionString=
  "E:\BOOKSHARE\BOOKS\SSIS\SSIS_R2\SAMPLES\SAMPLEDATA\NewContacts.txt">
        <DTS:FlatFileColumns>
          <DTS:FlatFileColumn
            DTS:ColumnType="Delimited"
            DTS:ColumnDelimiter="_x0009_"
            DTS:MaximumWidth="50"
            DTS:DataType="129"
            DTS:TextQualified="True"
            DTS:ObjectName="FirstName"
            DTS:DTSID="{64661EFB-1C85-47C6-974D-BDFC7441D404}"
            DTS:CreationName="" />
          <DTS:FlatFileColumn
            DTS:ColumnType="Delimited"
            DTS:ColumnDelimiter="_x000D__x000A_"
            DTS:MaximumWidth="50"
            DTS:DataType="129"
            DTS:TextQualified="True"
            DTS:ObjectName="LastName"
            DTS:DTSID="{2DEA7754-8936-4BFB-822E-F0014DE3C9C0}"
            DTS:CreationName="" />
        </DTS:FlatFileColumns>
      </DTS:ConnectionManager>
    </DTS:ObjectData>
  </DTS:ConnectionManager>
</DTS:ConnectionManagers>
<DTS:Configurations>
  <DTS:Configuration
    DTS:ConfigurationString="SSISSAMPLESROOT"
    DTS:ConfigurationType="2"
    DTS:ConfigurationVariable=
    "\Package.Variables[SAMPLES::ROOTWD].Properties[Value]"
    DTS:CreationName=""
    DTS:DTSID="{90405045-3A91-43C2-B759-6C183C0E81A6}"
    DTS:ObjectName="Configure The Root Folder" />
  <DTS:Configuration
    DTS:ConfigurationString="SSISSAMPLESERVER"
    DTS:ConfigurationType="2"
    DTS:ConfigurationVariable=
```

```
        "\Package.Variables[SAMPLES::SERVERNAME].Properties[Value]"
        DTS:CreationName=""
        DTS:DTSID="{A9F64F24-15C2-4198-A308-F9124C48468D}"
        DTS:ObjectName="Configure The Sample Server Name" />
    <DTS:Configuration
        DTS:ConfigurationString="SSISSAMPLESWORKINGFOLDER"
        DTS:ConfigurationType="2"
        DTS:ConfigurationVariable=
        "\Package.Variables[SAMPLES::WORKINGFOLDER].Properties[Value]"
        DTS:CreationName=""
        DTS:DTSID="{1BAB0708-C962-4F9B-BFD6-EC434567ACAF}"
        DTS:ObjectName="Configure the Working Folder variable" />
  </DTS:Configurations>
  <DTS:Variables>
    <DTS:Variable
        DTS:CreationName=""
        DTS:DTSID="{E05C4384-FA46-4EDB-9E41-E0CDD6DD4DC6}"
        DTS:EvaluateAsExpression="True"
        DTS:Expression="@ROOTWD + &quot;\\&quot; + @SAMPLESUBFOLDER"
        DTS:IncludeInDebugDump="2345"
        DTS:Namespace="SAMPLES"
        DTS:ObjectName="CURRENTWD">
        <DTS:VariableValue
          DTS:DataType="8">E:\BOOKSHARE\BOOKS\SSIS\SSIS_R2\SAMPLES\S21-Advanced
        </DTS:VariableValue>
    </DTS:Variable>
    <DTS:Variable
        DTS:CreationName=""
        DTS:DTSID="{3BD365BA-FE06-4B5A-8100-69F27C71B74E}"
        DTS:IncludeInDebugDump="2345"
        DTS:Namespace="SAMPLES"
        DTS:ObjectName="ROOTWD">
        <DTS:VariableValue
          DTS:DataType="8">E:\BOOKSHARE\BOOKS\SSIS\SSIS_R2\SAMPLES
        </DTS:VariableValue>
    </DTS:Variable>
    <DTS:Variable
        DTS:CreationName=""
        DTS:DTSID="{6E7FB8C5-7F98-454E-87E0-D9BF0D0E06AD}"
        DTS:IncludeInDebugDump="2345"
        DTS:Namespace="SAMPLES"
        DTS:ObjectName="SAMPLESUBFOLDER">
        <DTS:VariableValue
          DTS:DataType="8">S21-Advanced</DTS:VariableValue>
    </DTS:Variable>
    <DTS:Variable
        DTS:CreationName=""
        DTS:DTSID="{5F9998D0-186D-4C44-A949-66C8477A3CE7}"
        DTS:IncludeInDebugDump="2345"
        DTS:Namespace="SAMPLES"
        DTS:ObjectName="SERVERNAME">
        <DTS:VariableValue
          DTS:DataType="8">.</DTS:VariableValue>
    </DTS:Variable>
    <DTS:Variable
        DTS:CreationName=""
        DTS:DTSID="{DC4065D8-A163-4893-BE71-FB0D146F3BA8}"
        DTS:IncludeInDebugDump="2345"
        DTS:Namespace="SAMPLES"
        DTS:ObjectName="WORKINGFOLDER">
        <DTS:VariableValue
          DTS:DataType="8">E:\BOOKSHARE\BOOKS\SSIS\SSIS_R2\SAMPLES\SCRATCH
        </DTS:VariableValue>
    </DTS:Variable>
  </DTS:Variables>
  <DTS:Executables>
    <DTS:Executable
        DTS:refId="Package\Data Flow Task"
        DTS:CreationName="SSIS.Pipeline.3"
        DTS:Description="Data Flow Task"
```

```
        DTS:DTSID="{7FAA584E-2F45-4786-8410-8E97F54B0E8D}"
        DTS:ExecutableType="SSIS.Pipeline.3"
        DTS:ExecValue="0"
        DTS:ExecValueType="3"
        DTS:LocaleID="-1"
        DTS:ObjectName="Data Flow Task"
        DTS:TaskContact="Performs high-performance data extraction,
        transformation and loading;Microsoft Corporation;
        Microsoft SQL Server v9; (C) 2004 Microsoft Corporation;
 All Rights Reserved;http://www.microsoft.com/sql/support/default.asp;1">
        <DTS:Variables />
        <DTS:ObjectData>
          <pipeline
            BLOBTempStoragePath=""
            bufferTempStoragePath=""
            engineThreads="5"
            id="0"
            runInOptimizedMode="false"
            version="1">
            <components>
              <component
                refId="Package\Data Flow Task\Fuzzy Lookup"
                componentClassID="{E4A5F949-EC93-45AB-8B36-B52936257EC2}"
                contactInfo="Fuzzy Lookup;Microsoft Corporation;
                Microsoft SqlServer v9; (C) 2005 Microsoft Corporation;
                All Rights Reserved; http://www.microsoft.com/sql/support;5"
                description=
        "Looks up values in a reference dataset by using fuzzy matching."
                name="Fuzzy Lookup"
                version="5">
                <properties>
                  <property
                    dataType="System.Int32"
description="Indicates whether to reuse an existing fuzzy-match index
table or build a new one. The new index table can be temporary,
persisted, or persisted and maintained."
                    name="MatchIndexOptions"
                    typeConverter="tcMatchIndexOptions">1</property>
                  <property
                    dataType="System.Null"
description="Specifies the maximum amount of memory (MB)
                    that this instance of the Fuzzy Lookup transformation can
                    use. If 0 is given, then the transform will dynamically
                    adjust it's memory usage based upon it's requirements and
                    available system memory."
                    expressionType="Notify"
                    name="MaxMemoryUsage" />
                  <property
                    dataType="System.String"
description="Specifies the name of the fuzzy-match index
                    table to be reused or the table name in which a newly
                    generated fuzzy-match index table will be written."
                    name="MatchIndexName"
                    state="escaped">ContactsMatchIndex</property>
                  <property
                    dataType="System.String"
description="Specifies the name of the reference table
                    or view against which to match incoming records."
                    name="ReferenceTableName"
                    state="escaped">_x005B_Person_x005D_._x005B_Contact_x005D_
                  </property>
                  <property
                    dataType="System.Int32"
description="Indicates whether to drop an existing
                    fuzzy-match index table if transformation is configured to
                    generate a new table and a table with the same name already
                    exists."
                    name="DropExistingMatchIndex">1</property>
                  <property
                    dataType="System.Int32"
```

```
                  description="Whether a copy of the reference table should
                               be made for fuzzy match index construction and subsequent
                               matching.  It is strongly recommended that a copy always
                               be made, otherwise unpredicatable output may result if the
                               contents of the reference table change during package
                               execution (or between executions if reusing a fuzzy
                               match index).  If this property is set to false, the
                               reference table must additionally have a clustered index
                               built upon an integer identity column."
                               name="CopyReferenceTable">1</property>
                       <property
                       dataType="System.UInt32"
                  description="Specifies the maximum number of fuzzy matches
                               to return for an input record."
                               name="MaxOutputMatchesPerInput">1</property>
                       <property
                       dataType="System.Single"
                  description="Specifies the minimum similarity score
                               required for the match results that the transformation
                               returns."
                               name="MinSimilarity">0.65</property>
                       <property
                       dataType="System.String"
                  description="Specifies the set of white-space and
                               punctuation characters used to separate fuzzy match
                               fields into the words used for matching and scoring a
                               fuzzy lookup."
                               name="Delimiters"
                               state="escaped">
_x0020__x0009__x000D__x000A__x002C_._x003B_:-_x0022__x0027__x0026__x002F__
x005C__x0040__x0021__x003F__x0028__x0029__x003C__x003E__x005B__x005D__x007B
__x007D__x007C__x0023__x002A__x005E__x0025_</property>
                       <property
                       dataType="System.String"
                  description="Specifies the metadata of the reference
                               table in XML format."
                               name="ReferenceMetadataXml"
                               state="escaped">_x003C_referenceMetadata_x003E__x003C_
referenceColumns_x003E__x003C_referenceColumn_x0020_name_x003D__x0022_
ContactID_x0022__x0020_dataType_x003D__x0022_DT_I4_x0022__x0020_length_x003D
__x0022_0_x0022__x0020_precision_x003D__x0022_0_x0022__x0020_scale_x003D__
x0022_0_x0022__x0020_codePage_x003D__x0022_0_x0022__x002F__x003E__x003C_
referenceColumn_x0020_name_x003D__x0022_NameStyle_x0022__x0020_dataType_
x003D__x0022_DT_BOOL_x0022__x0020_length_x003D__x0022_0_x0022__x0020_
precision_x003D__x0022_0_x0022__x0020_scale_x003D__x0022_0_x0022__x0020_
codePage_x003D__x0022_0_x0022__x002F__x003E__x003C_referenceColumn_x0020_
name_x003D__x0022_Title_x0022__x0020_dataType_x003D__x0022_DT_WSTR_x0022__
x0020_length_x003D__x0022_8_x0022__x0020_precision_x003D__x0022_0_x0022__
x0020_scale_x003D__x0022_0_x0022__x0020_codePage_x003D__x0022_0_x0022__
x002F__x003E__x003C_referenceColumn_x0020_name_x003D__x0022_FirstName_x0022
__x0020_dataType_x003D__x0022_DT_WSTR_x0022__x0020_length_x003D__x0022_50_
x0022__x0020_precision_x003D__x0022_0_x0022__x0020_scale_x003D__x0022_0_
x0022__x0020_codePage_x003D__x0022_0_x0022__x002F__x003E__x003C_
referenceColumn_x0020_name_x003D__x0022_MiddleName_x0022__x0020_dataType_
x003D__x0022_DT_WSTR_x0022__x0020_length_x003D__x0022_50_x0022__x0020_
precision_x003D__x0022_0_x0022__x0020_scale_x003D__x0022_0_x0022__x0020_
codePage_x003D__x0022_0_x0022__x002F__x003E__x003C_referenceColumn_x0020_
name_x003D__x0022_LastName_x0022__x0020_dataType_x003D__x0022_DT_WSTR_x0022
__x0020_length_x003D__x0022_50_x0022__x0020_precision_x003D__x0022_0_x0022__
x0020_scale_x003D__x0022_0_x0022__x0020_codePage_x003D__x0022_0_x0022__x002F
__x003E__x003C_referenceColumn_x0020_name_x003D__x0022_Suffix_x0022__x0020
_dataType_x003D__x0022_DT_WSTR_x0022__x0020_length_x003D__x0022_10_x0022__
x0020_precision_x003D__x0022_0_x0022__x0020_scale_x003D__x0022_0_x0022__
x0020_codePage_x003D__x0022_0_x0022__x002F__x003E__x003C_referenceColumn_
x0020_name_x003D__x0022_EmailAddress_x0022__x0020_dataType_x003D__x0022_DT_
WSTR_x0022__x0020_length_x003D__x0022_50_x0022__x0020_precision_x003D__
x0022_0_x0022__x0020_scale_x003D__x0022_0_x0022__x0020_codePage_x003D__x0022
_0_x0022__x002F__x003E__x003C_referenceColumn_x0020_name_x003D__x0022_
EmailPromotion_x0022__x0020_dataType_x003D__x0022_DT_I4_x0022__x0020_length
_x003D__x0022_0_x0022__x0020_precision_x003D__x0022_0_x0022__x0020_scale_
```

```
x003D__x0022_0_x0022__x0020_codePage_x003D__x0022_0_x0022__x002F__x003E__
x003C_referenceColumn_x0020_name_x003D__x0022_Phone_x0022__x0020_dataType_
x003D__x0022_DT_WSTR_x0022__x0020_length_x003D__x0022_25_x0022__x0020_
precision_x003D__x0022_0_x0022__x0020_scale_x003D__x0022_0_x0022__x0020_
codePage_x003D__x0022_0_x0022__x002F__x003E__x003C_referenceColumn_x0020_
name_x003D__x0022_PasswordHash_x0022__x0020_dataType_x003D__x0022_DT_STR_
x0022__x0020_length_x003D__x0022_40_x0022__x0020_precision_x003D__x0022_0_
x0022__x0020_scale_x003D__x0022_0_x0022__x0020_codePage_x003D__x0022_1252_
x0022__x002F__x003E__x003C_referenceColumn_x0020_name_x003D__x0022_
PasswordSalt_x0022__x0020_dataType_x003D__x0022_DT_STR_x0022__x0020_
length_x003D__x0022_10_x0022__x0020_precision_x003D__x0022_0_x0022__
x0020_scale_x003D__x0022_0_x0022__x0020_codePage_x003D__x0022_1252_x0022__
x002F__x003E__x003C_referenceColumn_x0020_name_x003D__x0022_
AdditionalContactInfo_x0022__x0020_dataType_x003D__x0022_DT_NTEXT_x0022__
x0020_length_x003D__x0022_0_x0022__x0020_precision_x003D__x0022_0_x0022__
x0020_scale_x003D__x0022_0_x0022__x0020_codePage_x003D__x0022_0_x0022__
x002F__x003E__x003C_referenceColumn_x0020_name_x003D__x0022_rowguid_x0022__
x0020_dataType_x003D__x0022_DT_GUID_x0022__x0020_length_x003D__x0022_0_x0022
__x0020_precision_x003D__x0022_0_x0022__x0020_scale_x003D__x0022_0_x0022__
x0020_codePage_x003D__x0022_0_x0022__x002F__x003E__x003C_referenceColumn_
x0020_name_x003D__x0022_ModifiedDate_x0022__x0020_dataType_x003D__x0022_DT_
DBTIMESTAMP_x0022__x0020_length_x003D__x0022_0_x0022__x0020_precision_x003D
__x0022_0_x0022__x0020_scale_x003D__x0022_0_x0022__x0020_codePage_x003D__
x0022_0_x0022__x002F__x003E__x003C__x002F_referenceColumns_x003E__x003C__
x002F_referenceMetadata_x003E__x000D__x000A_</property>
                <property
                  dataType="System.Boolean"
    description="If true, each input record will be directly compared
    against EVERY record in the reference table.  This may produce slightly
    better results, but will be extremely slow unless the number of records
    in the reference table is very small.  Note that the entire reference
    table will be loaded into main memory if this option is enabled.
    This option is primarily intended for debugging purposes and should
    be used with care."
                  name="Exhaustive">false</property>
                <property
                  dataType="System.Boolean"
    description="If true, Fuzzy Lookup will partially load the error-tolerant
    index and the reference table into main memory before execution begins.
    This can greatly enhance performance if many input rows are to be
    processed."
                  name="WarmCaches">true</property>
              </properties>
              <connections>
                <connection
                  refId=
    "Package\Data Flow Task\Fuzzy Lookup.Connections[OleDbConnection]"
                  connectionManagerID=
    "Package.ConnectionManagers[.\SQL2K5.AdventureWorks]"
    description="Microsoft SQL Server connection and database in which the
    fuzzy lookup reference table is located and where the fuzzy match
    index table will be stored and accessed."
                  name="OleDbConnection" />
              </connections>
              <inputs>
                <input
  refId="Package\Data Flow Task\Fuzzy Lookup.Inputs[Fuzzy Lookup Input]"
                  name="Fuzzy Lookup Input">
                  <inputColumns>
                    <inputColumn
                      refId="Package\Data Flow Task\Fuzzy Lookup.Inputs
                      [Fuzzy Lookup Input].Columns[FirstName]"
                      cachedDataType="wstr"
                      cachedLength="50"
                      cachedName="FirstName"
          lineageId="Package\Data Flow Task\New Contact Names.Outputs
                      [Flat File Source Output].Columns[FirstName]">
                      <properties>
                        <property
                          dataType="System.String"
```

```
                     description="Name of the reference table column with which the input
column should be matched."
                                  name="JoinToReferenceColumn"
                                  state="escaped">FirstName</property>
                               <property
                                  dataType="System.Int32"
description="Specifies whether the match is a fuzzy or an exact match."
                                  name="JoinType"
                                  typeConverter="tcJoinType">2</property>
                               <property
                                  dataType="System.Single"
description="Specifies the minimum textual similarity required between
input and fuzzy matched records on values in this particular join column.
Must be a floating point value between 0.0 and 1.0."
                                  name="MinSimilarity">0</property>
                               <property
                                  dataType="System.Int32"
description="Specifies the comparison options for character data.
The options are those supported by the FoldString function."
                                  name="FuzzyComparisonFlags"
                                  typeConverter="tcComparisonFlags">1</property>
                               <property
                                  dataType="System.UInt32"
description="Flags to FoldString that should be used when comparing
whether two strings are equivalent."
                                  name="FuzzyComparisonFlagsEx"
                                  typeConverter="tcComparisonFlagsEx">0</property>
                            </properties>
                         </inputColumn>
                         <inputColumn
refId="Package\Data Flow Task\Fuzzy Lookup.Inputs
[Fuzzy Lookup Input].Columns[LastName]"
                            cachedDataType="wstr"
                            cachedLength="50"
                            cachedName="LastName"
lineageId="Package\Data Flow Task\New Contact Names.Outputs
[Flat File Source Output].Columns[LastName]">
                            <properties>
                               <property
                                  dataType="System.String"
 description="Name of the reference table column with which the
 input column should be matched."
                                  name="JoinToReferenceColumn"
                                  state="escaped">LastName</property>
                               <property
                                  dataType="System.Int32"
 description="Specifies whether the match is a fuzzy or an exact match."
                                  name="JoinType"
                                  typeConverter="tcJoinType">2</property>
                               <property
                                  dataType="System.Single"
 description="Specifies the minimum textual similarity required between
 input and fuzzy matched records on values in this particular join column.
 Must be a floating point value between 0.0 and 1.0."
                                  name="MinSimilarity">0</property>
                               <property
                                  dataType="System.Int32"
description="Specifies the comparison options for character data.
The options are those supported by the FoldString function."
                                  name="FuzzyComparisonFlags"
                                  typeConverter="tcComparisonFlags">1</property>
                               <property
                                  dataType="System.UInt32"
description="Flags to FoldString that should be used when comparing
whether two strings are equivalent."
                                  name="FuzzyComparisonFlagsEx"
                                  typeConverter="tcComparisonFlagsEx">0</property>
                            </properties>
                         </inputColumn>
                      </inputColumns>
```

```
                      <externalMetadataColumns />
                  </input>
              </inputs>
              <outputs>
                  <output
  refId="Package\Data Flow Task\Fuzzy Lookup.Outputs[Fuzzy Lookup Output]"
                  name="Fuzzy Lookup Output">
                    <outputColumns>
                      <outputColumn
  refId="Package\Data Flow Task\Fuzzy Lookup.Outputs[Fuzzy Lookup Output].
  Columns[FirstName]"
                          dataType="wstr"
                          length="50"
  lineageId="Package\Data Flow Task\Fuzzy Lookup.Outputs
  [Fuzzy Lookup Output].Columns[FirstName]"
                          name="FirstName">
                        <properties>
                          <property
                            dataType="System.Null"
  description="Specifies the reference table column that is copied to the
  output for each match result."
                            name="CopyFromReferenceColumn"
                            state="escaped" />
                          <property
                            containsID="true"
                            dataType="System.Int32"
  description="Specifies the lineage identifier of the input column from
  which to copy to this output column for each match."
                            name="SourceInputColumnLineageID">
#{Package\Data Flow Task\New Contact Names.Outputs[Flat File Source Output]
.Columns[FirstName]}</property>
                          <property
                            dataType="System.Int32"
  description="Specifies whether the column is used to indicate confidence
  or similarity."
                            name="ColumnType"
                            typeConverter="tcColumnType">0</property>
                        </properties>
                      </outputColumn>
                      <outputColumn
  refId="Package\Data Flow Task\Fuzzy Lookup.Outputs[Fuzzy Lookup Output].
  Columns[LastName]"
                          dataType="wstr"
                          length="50"
                          lineageId="Package\Data Flow Task\Fuzzy Lookup.
                          Outputs[Fuzzy Lookup Output].Columns[LastName]"
                          name="LastName">
                        <properties>
                          <property
                            dataType="System.Null"
description="Specifies the reference table column that is copied to the
output for each match result."
                            name="CopyFromReferenceColumn"
                            state="escaped" />
                          <property
                            containsID="true"
                            dataType="System.Int32"
  description="Specifies the lineage identifier of the input column from
  which to copy to this output column for each match."
                            name="SourceInputColumnLineageID">
  #{Package\Data Flow Task\New Contact Names.Outputs[Flat File Source
  Output].Columns[LastName]}</property>
                          <property
                            dataType="System.Int32"
  description="Specifies whether the column is used to indicate
  confidence or similarity."
                            name="ColumnType"
                            typeConverter="tcColumnType">0</property>
                        </properties>
                      </outputColumn>
```

```
                    <outputColumn
                      refId="Package\Data Flow Task\Fuzzy Lookup.Outputs
                      [Fuzzy Lookup Output].Columns[FLFirstName]"
                      dataType="wstr"
                      length="50"
lineageId="Package\Data Flow Task\Fuzzy Lookup.Outputs[Fuzzy
Lookup Output].Columns[FLFirstName]"
                      name="FLFirstName">
                      <properties>
                        <property
                          dataType="System.String"
description="Specifies the reference table column that is copied to the
output for each match result."
                          name="CopyFromReferenceColumn"
                          state="escaped">FirstName</property>
                        <property
                          containsID="true"
                          dataType="System.Null"
description="Specifies the lineage identifier of the input column
from which to copy to this output column for each match."
                          name="SourceInputColumnLineageID" />
                        <property
                          dataType="System.Int32"
description="Specifies whether the column is used to indicate
confidence or similarity."
                          name="ColumnType"
                          typeConverter="tcColumnType">0</property>
                      </properties>
                    </outputColumn>
                    <outputColumn
refId="Package\Data Flow Task\Fuzzy Lookup.Outputs
[Fuzzy Lookup Output].Columns[FLLastName]"
                      dataType="wstr"
                      length="50"
                      lineageId="Package\Data Flow Task\Fuzzy Lookup.Outputs
                      [Fuzzy Lookup Output].Columns[FLLastName]"
                      name="FLLastName">
                      <properties>
                        <property
                          dataType="System.String"
description="Specifies the reference table column that is copied to
the output for each match result."
                          name="CopyFromReferenceColumn"
                          state="escaped">LastName</property>
                        <property
                          containsID="true"
                          dataType="System.Null"
description="Specifies the lineage identifier of the input column from
which to copy to this output column for each match."
                          name="SourceInputColumnLineageID" />
                        <property
                          dataType="System.Int32"
description="Specifies whether the column is used to indicate
confidence or similarity."
                          name="ColumnType"
                          typeConverter="tcColumnType">0</property>
                      </properties>
                    </outputColumn>
                    <outputColumn
                      refId="Package\Data Flow Task\Fuzzy Lookup.Outputs
                      [Fuzzy Lookup Output].Columns[_Similarity]"
                      dataType="r4"
                      lineageId="Package\Data Flow Task\Fuzzy Lookup.Outputs
                      [Fuzzy Lookup Output].Columns[_Similarity]"
                      name="_Similarity">
                      <properties>
                        <property
                          dataType="System.Int32"
description="Specifies whether the column is used to indicate confidence
or similarity."
```

```
                          name="ColumnType"
                          typeConverter="tcColumnType">1</property>
                     </properties>
                  </outputColumn>
                  <outputColumn
                    refId="Package\Data Flow Task\Fuzzy Lookup.Outputs
                    [Fuzzy Lookup Output].Columns[_Confidence]"
                    dataType="r4"
                    lineageId="Package\Data Flow Task\Fuzzy Lookup.Outputs
                    [Fuzzy Lookup Output].Columns[_Confidence]"
                    name="_Confidence">
                     <properties>
                       <property
                         dataType="System.Int32"
      description="Specifies whether the column is used to indicate confidence
      or similarity."
                          name="ColumnType"
                          typeConverter="tcColumnType">2</property>
                     </properties>
                  </outputColumn>
                  <outputColumn
                    refId="Package\Data Flow Task\Fuzzy Lookup.Outputs
                    [Fuzzy Lookup Output].Columns[_Similarity_FirstName]"
                    dataType="r4"
                    description="Column-level similarity score for fuzzy
                    join on column &quot;FirstName&quot;."
                    lineageId="Package\Data Flow Task\Fuzzy Lookup.Outputs
                    [Fuzzy Lookup Output].Columns[_Similarity_FirstName]"
                    name="_Similarity_FirstName">
                     <properties>
                       <property
                         dataType="System.Null"
      description="Specifies the reference table column that is copied to the
      output for each match result."
                          name="CopyFromReferenceColumn"
                          state="escaped" />
                       <property
                         containsID="true"
                         dataType="System.Int32"
      description="Specifies the lineage identifier of the input column from
      which to copy to this output column for each match."
                          name="SourceInputColumnLineageID">
      #{Package\Data Flow Task\New Contact Names.Outputs[Flat File Source Output]
      .Columns[FirstName]}</property>
                       <property
                         dataType="System.Int32"
      description="Specifies whether the column is used to indicate confidence
      or similarity."
                          name="ColumnType"
                          typeConverter="tcColumnType">3</property>
                     </properties>
                  </outputColumn>
                  <outputColumn
                    refId="Package\Data Flow Task\Fuzzy Lookup.Outputs
                    [Fuzzy Lookup Output].Columns[_Similarity_LastName]"
                    dataType="r4"
                    description="Column-level similarity score for fuzzy
                    join on column &quot;LastName&quot;."
                    lineageId="Package\Data Flow Task\Fuzzy Lookup.Outputs
                    [Fuzzy Lookup Output].Columns[_Similarity_LastName]"
                    name="_Similarity_LastName">
                     <properties>
                       <property
                         dataType="System.Null"
      description="Specifies the reference table column that is copied to the
      output for each match result."
                          name="CopyFromReferenceColumn"
                          state="escaped" />
                       <property
                         containsID="true"
```

```
                              dataType="System.Int32"
  description="Specifies the lineage identifier of the input column from
  which to copy to this output column for each match."
                              name="SourceInputColumnLineageID">
#{Package\Data Flow Task\New Contact Names.Outputs[Flat File Source Output].
Columns[LastName]}</property>
                            <property
                              dataType="System.Int32"
  description="Specifies whether the column is used to indicate confidence
  or similarity."
                              name="ColumnType"
                              typeConverter="tcColumnType">3</property>
                          </properties>
                        </outputColumn>
                      </outputColumns>
                      <externalMetadataColumns />
                    </output>
                  </outputs>
                </component>
                <component
                  refId="Package\Data Flow Task\Lookup"
                  componentClassID="{671046B0-AA63-4C9F-90E4-C06E0B710CE3}"
                  contactInfo="Lookup;Microsoft Corporation;Microsoft SqlServer
                  v9; (C) 2005 Microsoft Corporation; All Rights Reserved;
                  http://www.microsoft.com/sql/support;4"
 description="Looks up values in a reference dataset by using exact matching."
                  name="Lookup"
                  usesDispositions="true"
                  version="6">
                  <properties>
                    <property
                      dataType="System.String"
  description="Specifies the SQL statement that generates the lookup table."
                      expressionType="Notify"
                      name="SqlCommand"
 UITypeEditor="Microsoft.DataTransformationServices.Controls.
 ModalMultilineStringEditor, Microsoft.DataTransformationServices.Controls,
 Version=11.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91">
                      select * from [Person].[Contact]</property>
                    <property
                      dataType="System.String"
  description="Specifies a SQL statement that uses parameters to generate
  the lookup table."
                      expressionType="Notify"
                      name="SqlCommandParam">select * from (select * from
                      [Person].[Contact]) [refTable] where [refTable].
                      [FirstName] = ? and [refTable].[LastName] = ?
                    </property>
                    <property
                      dataType="System.Int32"
    description="Specifies the cache type of the lookup table."
                      name="CacheType"
                      typeConverter="CacheType">1</property>
                    <property
                      dataType="System.Int32"
  description="Maximum Memory Usage for Reference Cache on a 32 bit
  platform."
                      name="MaxMemoryUsage">25</property>
                    <property
                      dataType="System.Int64"
  description="Maximum Memory Usage for Reference Cache on a 64 bit
  platform."
                      name="MaxMemoryUsage64">25</property>
                    <property
                      dataType="System.String"
  description="Indicates whether to reference metadata in an XML format."
                      name="ReferenceMetadataXml">
&lt;referenceMetadata&gt;&lt;referenceColumns&gt;&lt;referenceColumn name=
"ContactID" dataType="DT_I4" length="0" precision="0" scale="0" codePage="0"
/&gt;&lt;referenceColumn name="NameStyle" dataType="DT_BOOL" length="0"
```

```
precision="0" scale="0" codePage="0"/&gt;&lt;referenceColumn name="Title"
dataType="DT_WSTR" length="8" precision="0" scale="0" codePage="0"/&gt;&lt;
referenceColumn name="FirstName" dataType="DT_WSTR" length="50" precision=
"0" scale="0" codePage="0"/&gt;&lt;referenceColumn name="MiddleName"
dataType="DT_WSTR" length="50" precision="0" scale="0" codePage="0"/&gt;&lt;
referenceColumn name="LastName" dataType="DT_WSTR" length="50" precision="0"
scale="0" codePage="0"/&gt;&lt;referenceColumn name="Suffix" dataType=
"DT_WSTR" length="10" precision="0" scale="0" codePage="0"/&gt;&lt;
referenceColumn name="EmailAddress" dataType="DT_WSTR" length="50"
precision="0" scale="0" codePage="0"/&gt;&lt;referenceColumn name=
"EmailPromotion" dataType="DT_I4" length="0" precision="0" scale="0"
codePage="0"/&gt;&lt;referenceColumn name="Phone" dataType="DT_WSTR"
length="25" precision="0" scale="0" codePage="0"/&gt;&lt;referenceColumn
name="PasswordHash" dataType="DT_STR" length="128" precision="0" scale="0"
codePage="1252"/&gt;&lt;referenceColumn name="PasswordSalt" dataType=
"DT_STR" length="10" precision="0" scale="0" codePage="1252"/&gt;&lt;
referenceColumn name="AdditionalContactInfo" dataType="DT_NTEXT" length="0"
precision="0" scale="0" codePage="0"/&gt;&lt;referenceColumn name="rowguid"
dataType="DT_GUID" length="0" precision="0" scale="0" codePage="0"/&gt;&lt;
referenceColumn name="ModifiedDate" dataType="DT_DBTIMESTAMP" length="0"
precision="0" scale="0" codePage="0"/&gt;&lt;/referenceColumns&gt;&lt;
/referenceMetadata&gt;</property>
                <property
                  containsID="true"
                  dataType="System.String"
  description="Specifies the list of lineage identifiers that map to the
  parameters that the SQL statement in the SQLCommand property uses.
  Entries in the list are separated by semicolons."
  name="ParameterMap">#{Package\Data Flow Task\New Contact Names.Outputs
  [Flat File Source Output].Columns[FirstName]};#{Package\Data Flow Task\
  New Contact Names.Outputs[Flat File Source Output].Columns[LastName]};
</property>
                <property
                  dataType="System.Int32"
  description="Specifies the column code page to use when code page
  information is unavailable from the data source."
                  name="DefaultCodePage">1252</property>
                <property
                  dataType="System.Int32"
  description="Specifies the type of connection used to access the
  reference dataset."
                  name="ConnectionType"
                  typeConverter="LookupConnectionType">0</property>
                <property
                  dataType="System.Int32"
  description="Specifies how the Lookup transformation handles rows without
  matching entries in the reference data set."
                  name="NoMatchBehavior"
                  typeConverter="LookupNoMatchBehavior">0</property>
                <property
                  dataType="System.Int32"
  description="Specifies the percentage of the cache that is allocated for
  rows with no matching entries in the reference dataset."
                  name="NoMatchCachePercentage">0</property>
                <property
                  dataType="System.Boolean"
  description="Determines whether duplicate keys in the reference data
  should be treated as errors when full cache mode is used."
                  name="TreatDuplicateKeysAsError">false</property>
              </properties>
              <connections>
                <connection
  refId="Package\Data Flow Task\Lookup.Connections[OleDbConnection]"
                  connectionManagerID="Package.ConnectionManagers
                  [.\SQL2K5.AdventureWorks]"
  description="Connection manager used to access lookup data."
                  name="OleDbConnection" />
              </connections>
              <inputs>
                <input
```

```
                    refId="Package\Data Flow Task\Lookup.Inputs
                    [Lookup Input]"
                    name="Lookup Input">
                    <inputColumns>
                      <inputColumn
                        refId="Package\Data Flow Task\Lookup.Inputs
                        [Lookup Input].Columns[FirstName]"
                        cachedDataType="wstr"
                        cachedLength="50"
                        cachedName="FirstName"
lineageId="Package\Data Flow Task\New Contact Names.Outputs
[Flat File Source Output].Columns[FirstName]">
                        <properties>
                          <property
                            dataType="System.String"
description="Specifies the column in the reference table that
a column joins."
                            name="JoinToReferenceColumn">
                            FirstName</property>
                          <property
                            dataType="System.Null"
description="Specifies the column in the reference table from
which a column is copied."
                            name="CopyFromReferenceColumn" />
                        </properties>
                      </inputColumn>
                      <inputColumn
refId="Package\Data Flow Task\Lookup.Inputs[Lookup Input].
Columns[LastName]"
                        cachedDataType="wstr"
                        cachedLength="50"
                        cachedName="LastName"
lineageId="Package\Data Flow Task\New Contact Names.Outputs
[Flat File Source Output].Columns[LastName]">
                        <properties>
                          <property
                            dataType="System.String"
description="Specifies the column in the reference table that
a column joins."
                            name="JoinToReferenceColumn">LastName</property>
                          <property
                            dataType="System.Null"
description="Specifies the column in the reference table from which
a column is copied."
                            name="CopyFromReferenceColumn" />
                        </properties>
                      </inputColumn>
                    </inputColumns>
                    <externalMetadataColumns />
                  </input>
                </inputs>
                <outputs>
                  <output
 refId="Package\Data Flow Task\Lookup.Outputs[Lookup Match Output]"
                    errorOrTruncationOperation="Lookup"
                    errorRowDisposition="RedirectRow"
                    exclusionGroup="1"
                    name="Lookup Match Output"
  synchronousInputId="Package\Data Flow Task\Lookup.
  Inputs[Lookup Input]">
                    <outputColumns>
                      <outputColumn
  refId="Package\Data Flow Task\Lookup.Outputs[Lookup Match Output].
  Columns[LUFirstName]"
                        dataType="wstr"
                        errorOrTruncationOperation="Copy Column"
                        length="50"
                        lineageId="Package\Data Flow Task\Lookup.Outputs
                        [Lookup Match Output].Columns[LUFirstName]"
                        name="LUFirstName"
```

```
                          truncationRowDisposition="FailComponent">
                          <properties>
                            <property
                              dataType="System.String"
description="Specifies the column in the reference table from
which a column is copied."
                              name="CopyFromReferenceColumn">FirstName</property>
                          </properties>
                        </outputColumn>
                        <outputColumn
                          refId="Package\Data Flow Task\Lookup.Outputs
                          [Lookup Match Output].Columns[LULastName]"
                          dataType="wstr"
                          errorOrTruncationOperation="Copy Column"
                          length="50"
                          lineageId="Package\Data Flow Task\Lookup.Outputs
                          [Lookup Match Output].Columns[LULastName]"
                          name="LULastName"
                          truncationRowDisposition="FailComponent">
                          <properties>
                            <property
                              dataType="System.String"
description="Specifies the column in the reference table from
which a column is copied."
                              name="CopyFromReferenceColumn">LastName</property>
                          </properties>
                        </outputColumn>
                        <outputColumn
                          refId="Package\Data Flow Task\Lookup.Outputs
                          [Lookup Match Output].Columns[NameStyle]"
                          dataType="bool"
                          errorOrTruncationOperation="Copy Column"
                          lineageId="Package\Data Flow Task\Lookup.Outputs
                          [Lookup Match Output].Columns[NameStyle]"
                          name="NameStyle"
                          truncationRowDisposition="FailComponent">
                          <properties>
                            <property
                              dataType="System.String"
description="Specifies the column in the reference table from
which a column is copied."
                              name="CopyFromReferenceColumn">NameStyle
                            </property>
                          </properties>
                        </outputColumn>
                        <outputColumn
                          refId="Package\Data Flow Task\Lookup.Outputs
                          [Lookup Match Output].Columns[Title]"
                          dataType="wstr"
                          errorOrTruncationOperation="Copy Column"
                          length="8"
                          lineageId="Package\Data Flow Task\Lookup.Outputs
                          [Lookup Match Output].Columns[Title]"
                          name="Title"
                          truncationRowDisposition="FailComponent">
                          <properties>
                            <property
                              dataType="System.String"
description="Specifies the column in the reference table from
which a column is copied."
                              name="CopyFromReferenceColumn">Title</property>
                          </properties>
                        </outputColumn>
                        <outputColumn
                          refId="Package\Data Flow Task\Lookup.Outputs
                          [Lookup Match Output].Columns[MiddleName]"
                          dataType="wstr"
                          errorOrTruncationOperation="Copy Column"
                          length="50"
                          lineageId="Package\Data Flow Task\Lookup.Outputs
```

```
                      [Lookup Match Output].Columns[MiddleName]"
                      name="MiddleName"
                      truncationRowDisposition="FailComponent">
                      <properties>
                        <property
                          dataType="System.String"
description="Specifies the column in the reference table
from which a column is copied."
                    name="CopyFromReferenceColumn">MiddleName</property>
                      </properties>
                    </outputColumn>
                    <outputColumn
                      refId="Package\Data Flow Task\Lookup.Outputs
                      [Lookup Match Output].Columns[Suffix]"
                      dataType="wstr"
                      errorOrTruncationOperation="Copy Column"
                      length="10"
                      lineageId="Package\Data Flow Task\Lookup.
                      Outputs[Lookup Match Output].Columns[Suffix]"
                      name="Suffix"
                      truncationRowDisposition="FailComponent">
                      <properties>
                        <property
                          dataType="System.String"
description="Specifies the column in the reference table
from which a column is copied."
                    name="CopyFromReferenceColumn">Suffix</property>
                      </properties>
                    </outputColumn>
                    <outputColumn
                      refId="Package\Data Flow Task\Lookup.Outputs
                      [Lookup Match Output].Columns[EmailAddress]"
                      dataType="wstr"
                      errorOrTruncationOperation="Copy Column"
                      length="50"
                      lineageId="Package\Data Flow Task\Lookup.Outputs
                      [Lookup Match Output].Columns[EmailAddress]"
                      name="EmailAddress"
                      truncationRowDisposition="FailComponent">
                      <properties>
                        <property
                          dataType="System.String"
description="Specifies the column in the reference table
from which a column is copied."
                  name="CopyFromReferenceColumn">EmailAddress</property>
                      </properties>
                    </outputColumn>
                    <outputColumn
                      refId="Package\Data Flow Task\Lookup.Outputs
                      [Lookup Match Output].Columns[EmailPromotion]"
                      dataType="i4"
                      errorOrTruncationOperation="Copy Column"
                      lineageId="Package\Data Flow Task\Lookup.Outputs
                      [Lookup Match Output].Columns[EmailPromotion]"
                      name="EmailPromotion"
                      truncationRowDisposition="FailComponent">
                      <properties>
                        <property
                          dataType="System.String"
description="Specifies the column in the reference table from which
a column is copied."
                  name="CopyFromReferenceColumn">EmailPromotion</property>
                      </properties>
                    </outputColumn>
                    <outputColumn
                      refId="Package\Data Flow Task\Lookup.Outputs
                      [Lookup Match Output].Columns[Phone]"
                      dataType="wstr"
                      errorOrTruncationOperation="Copy Column"
                      length="25"
```

```
                          lineageId="Package\Data Flow Task\Lookup.Outputs
                          [Lookup Match Output].Columns[Phone]"
                          name="Phone"
                          truncationRowDisposition="FailComponent">
                          <properties>
                            <property
                              dataType="System.String"
description="Specifies the column in the reference table
from which a column is copied."
                              name="CopyFromReferenceColumn">Phone</property>
                          </properties>
                        </outputColumn>
                        <outputColumn
                          refId="Package\Data Flow Task\Lookup.Outputs
                          [Lookup Match Output].Columns[PasswordHash]"
                          codePage="1252"
                          dataType="str"
                          errorOrTruncationOperation="Copy Column"
                          length="128"
                          lineageId="Package\Data Flow Task\Lookup.Outputs
                          [Lookup Match Output].Columns[PasswordHash]"
                          name="PasswordHash"
                          truncationRowDisposition="FailComponent">
                          <properties>
                            <property
                              dataType="System.String"
description="Specifies the column in the reference table
from which a column is copied."
                              name="CopyFromReferenceColumn">PasswordHash</property>
                          </properties>
                        </outputColumn>
                        <outputColumn
                          refId="Package\Data Flow Task\Lookup.Outputs
                          [Lookup Match Output].Columns[PasswordSalt]"
                          codePage="1252"
                          dataType="str"
                          errorOrTruncationOperation="Copy Column"
                          length="10"
                          lineageId="Package\Data Flow Task\Lookup.Outputs
                          [Lookup Match Output].Columns[PasswordSalt]"
                          name="PasswordSalt"
                          truncationRowDisposition="FailComponent">
                          <properties>
                            <property
                              dataType="System.String"
description="Specifies the column in the reference table
from which a column is copied."
                name="CopyFromReferenceColumn">PasswordSalt</property>
                          </properties>
                        </outputColumn>
                        <outputColumn
                          refId="Package\Data Flow Task\Lookup.Outputs
                          [Lookup Match Output].Columns[AdditionalContactInfo]"
                          dataType="nText"
                          errorOrTruncationOperation="Copy Column"
                          lineageId="Package\Data Flow Task\Lookup.Outputs
                          [Lookup Match Output].Columns[AdditionalContactInfo]"
                          name="AdditionalContactInfo"
                          truncationRowDisposition="FailComponent">
                          <properties>
                            <property
                              dataType="System.String"
description="Specifies the column in the reference table
from which a column is copied."
name="CopyFromReferenceColumn">AdditionalContactInfo</property>
                          </properties>
                        </outputColumn>
                        <outputColumn
                          refId="Package\Data Flow Task\Lookup.Outputs
                          [Lookup Match Output].Columns[rowguid]"
```

```
                            dataType="guid"
                            errorOrTruncationOperation="Copy Column"
                            lineageId="Package\Data Flow Task\Lookup.Outputs
                            [Lookup Match Output].Columns[rowguid]"
                            name="rowguid"
                            truncationRowDisposition="FailComponent">
                            <properties>
                              <property
                                dataType="System.String"
    description="Specifies the column in the reference table
from which a column is copied."
                          name="CopyFromReferenceColumn">rowguid</property>
                            </properties>
                          </outputColumn>
                          <outputColumn
                            refId="Package\Data Flow Task\Lookup.Outputs
                            [Lookup Match Output].Columns[ModifiedDate]"
                            dataType="dbTimeStamp"
                            errorOrTruncationOperation="Copy Column"
                            lineageId="Package\Data Flow Task\Lookup.Outputs
                            [Lookup Match Output].Columns[ModifiedDate]"
                            name="ModifiedDate"
                            truncationRowDisposition="FailComponent">
                            <properties>
                              <property
                                dataType="System.String"
    description="Specifies the column in the reference table
from which a column is copied."
                          name="CopyFromReferenceColumn">ModifiedDate</property>
                            </properties>
                          </outputColumn>
                        </outputColumns>
                        <externalMetadataColumns />
                      </output>
                      <output
                        refId="Package\Data Flow Task\Lookup.Outputs
                        [Lookup No Match Output]"
    description="The Lookup output that handles rows with no matching entries
in the reference dataset. Use this output when the NoMatchBehavior
property is set to &quot;Send rows with no matching entries to the no
match output.&quot;"
                        exclusionGroup="1"
                        name="Lookup No Match Output"
      synchronousInputId="Package\Data Flow Task\Lookup.Inputs[Lookup Input]">
                        <externalMetadataColumns />
                      </output>
                      <output
    refId="Package\Data Flow Task\Lookup.Outputs[Lookup Error Output]"
                        exclusionGroup="1"
                        isErrorOut="true"
                        name="Lookup Error Output"
       synchronousInputId="Package\Data Flow Task\Lookup.Inputs[Lookup Input]">
                        <outputColumns>
                          <outputColumn
                            refId="Package\Data Flow Task\Lookup.Outputs
                            [Lookup Error Output].Columns[ErrorCode]"
                            dataType="i4"
                            lineageId="Package\Data Flow Task\Lookup.Outputs
                            [Lookup Error Output].Columns[ErrorCode]"
                            name="ErrorCode"
                            specialFlags="1" />
                          <outputColumn
                            refId="Package\Data Flow Task\Lookup.Outputs
                            [Lookup Error Output].Columns[ErrorColumn]"
                            dataType="i4"
                            lineageId="Package\Data Flow Task\Lookup.Outputs
                            [Lookup Error Output].Columns[ErrorColumn]"
                            name="ErrorColumn"
                            specialFlags="2" />
                        </outputColumns>
```

```
                  <externalMetadataColumns />
                </output>
              </outputs>
            </component>
            <component
              refId="Package\Data Flow Task\Multicast"
              componentClassID="{EC139FBC-694E-490B-8EA7-35690FB0F445}"
              contactInfo="Multicast;Microsoft Corporation;Microsoft
              SqlServer v9; (C) 2005 Microsoft Corporation; All Rights
              Reserved; http://www.microsoft.com/sql/support;0"
              description="Creates copies of a dataset."
              name="Multicast">
              <inputs>
                <input
        refId="Package\Data Flow Task\Multicast.Inputs[Multicast Input 1]"
                  name="Multicast Input 1">
                  <externalMetadataColumns />
                </input>
              </inputs>
              <outputs>
                <output
                  refId="Package\Data Flow Task\Multicast.Outputs
                  [Multicast Output 1]"
                  dangling="true"
                  deleteOutputOnPathDetached="true"
                  name="Multicast Output 1"
                  synchronousInputId="Package\Data Flow Task\Multicast.
                  Inputs[Multicast Input 1]">
                  <externalMetadataColumns />
                </output>
              </outputs>
            </component>
            <component
              refId="Package\Data Flow Task\Multicast 1"
              componentClassID="{EC139FBC-694E-490B-8EA7-35690FB0F445}"
              contactInfo="Multicast;Microsoft Corporation;Microsoft
              SqlServer v9; (C) 2005 Microsoft Corporation; All Rights
              Reserved; http://www.microsoft.com/sql/support;0"
              description="Creates copies of a dataset."
              name="Multicast 1">
              <inputs>
                <input
    refId="Package\Data Flow Task\Multicast 1.Inputs[Multicast Input 1]"
                  name="Multicast Input 1">
                  <externalMetadataColumns />
                </input>
              </inputs>
              <outputs>
                <output
    refId="Package\Data Flow Task\Multicast 1.Outputs[Multicast Output 2]"
                  dangling="true"
                  deleteOutputOnPathDetached="true"
                  name="Multicast Output 2"
                  synchronousInputId=
    "Package\Data Flow Task\Multicast 1.Inputs[Multicast Input 1]">
                  <externalMetadataColumns />
                </output>
              </outputs>
            </component>
            <component
              refId="Package\Data Flow Task\New Contact Names"
              componentClassID="{D23FD76B-F51D-420F-BBCB-19CBF6AC1AB4}"
              contactInfo="Flat File Source;Microsoft Corporation;Microsoft
              SqlServer v9; (C) 2005 Microsoft Corporation; All Rights
              Reserved; http://www.microsoft.com/sql/support;1"
  description="Flat File Source"
              localeId="1033"
              name="New Contact Names"
              usesDispositions="true"
              version="1">
```

```
                    <properties>
                      <property
                        dataType="System.Boolean"
description="Specifies whether zero-length columns are treated as null."
                        name="RetainNulls">false</property>
                      <property
                        dataType="System.String"
description="Specifies the name of an output column containing
the file name. If no name is specified, no output column containing
the file name will be generated."
                        name="FileNameColumnName"></property>
                    </properties>
                    <connections>
                      <connection
refId="Package\Data Flow Task\New Contact Names.
Connections[FlatFileConnection]"
connectionManagerID="Package.ConnectionManagers[New Contacts]"
                        name="FlatFileConnection" />
                    </connections>
                    <outputs>
                      <output
                        refId="Package\Data Flow Task\New Contact Names.
                        Outputs[Flat File Source Output]"
                        name="Flat File Source Output">
                        <outputColumns>
                          <outputColumn
                            refId="Package\Data Flow Task\New Contact Names.
                            Outputs[Flat File Source Output].Columns[FirstName]"
                            dataType="wstr"
                            errorOrTruncationOperation="Conversion"
                            errorRowDisposition="FailComponent"
                            externalMetadataColumnId="Package\Data Flow Task\
                            New Contact Names.Outputs[Flat File Source Output]
                            .ExternalColumns[FirstName]"
                            length="50"
                            lineageId="Package\Data Flow Task\New Contact Names.
                            Outputs[Flat File Source Output].Columns[FirstName]"
                            name="FirstName"
                            truncationRowDisposition="FailComponent">
                            <properties>
                              <property
                                dataType="System.Boolean"
description="Indicates whether the column uses the faster,
locale-neutral parsing routines."
                                name="FastParse">false</property>
                            </properties>
                          </outputColumn>
                          <outputColumn
                            refId="Package\Data Flow Task\New Contact Names.
                            Outputs[Flat File Source Output].Columns[LastName]"
                            dataType="wstr"
                            errorOrTruncationOperation="Conversion"
                            errorRowDisposition="FailComponent"
                            externalMetadataColumnId="Package\Data Flow Task\
                            New Contact Names.Outputs[Flat File Source Output].
                            ExternalColumns[LastName]"
                            length="50"
                            lineageId="Package\Data Flow Task\New Contact Names.
                            Outputs[Flat File Source Output].Columns[LastName]"
                            name="LastName"
                            truncationRowDisposition="FailComponent">
                            <properties>
                              <property
                                dataType="System.Boolean"
description="Indicates whether the column uses the faster,
locale-neutral parsing routines."
                                name="FastParse">false</property>
                            </properties>
                          </outputColumn>
                        </outputColumns>
```

```
                      <externalMetadataColumns
                        isUsed="True">
                        <externalMetadataColumn
refId="Package\Data Flow Task\New Contact Names.Outputs
[Flat File Source Output].ExternalColumns[FirstName]"
                          codePage="1252"
                          dataType="str"
                          length="50"
                          name="FirstName" />
                        <externalMetadataColumn
refId="Package\Data Flow Task\New Contact Names.Outputs
[Flat File Source Output].ExternalColumns[LastName]"
                          codePage="1252"
                          dataType="str"
                          length="50"
                          name="LastName" />
                      </externalMetadataColumns>
                    </output>
                    <output
 refId="Package\Data Flow Task\New Contact Names.Outputs
 [Flat File Source Error Output]"
                      isErrorOut="true"
                      name="Flat File Source Error Output">
                      <outputColumns>
                        <outputColumn
                          refId="Package\Data Flow Task\New Contact
                          Names.Outputs[Flat File Source Error Output].
                          Columns[Flat File Source Error Output Column]"
                          codePage="1252"
                          dataType="text"
                          description="Flat File Source Error Output Column"
                          lineageId="Package\Data Flow Task\New Contact Names.
                          Outputs[Flat File Source Error Output].Columns
                          [Flat File Source Error Output Column]"
                          name="Flat File Source Error Output Column" />
                        <outputColumn
                          refId="Package\Data Flow Task\New Contact Names.Outputs
                          [Flat File Source Error Output].Columns[ErrorCode]"
                          dataType="i4"
              lineageId="Package\Data Flow Task\New Contact Names.
              Outputs[Flat File Source Error Output].Columns[ErrorCode]"
                          name="ErrorCode"
                          specialFlags="1" />
                        <outputColumn
refId="Package\Data Flow Task\New Contact Names.Outputs
[Flat File Source Error Output].Columns[ErrorColumn]"
                          dataType="i4"
lineageId="Package\Data Flow Task\New Contact Names.Outputs
[Flat File Source Error Output].Columns[ErrorColumn]"
                          name="ErrorColumn"
                          specialFlags="2" />
                      </outputColumns>
                      <externalMetadataColumns />
                    </output>
                  </outputs>
                </component>
              </components>
              <paths>
                <path
                  refId="Package\Data Flow Task.Paths[Flat File Source Output]"
                  endId="Package\Data Flow Task\Lookup.Inputs[Lookup Input]"
                  name="Flat File Source Output"
            startId="Package\Data Flow Task\New Contact Names.Outputs
            [Flat File Source Output]" />
                <path
                  refId="Package\Data Flow Task.Paths[Fuzzy Lookup Output]"
                  endId="Package\Data Flow Task\Multicast 1.Inputs
                  [Multicast Input 1]"
                  name="Fuzzy Lookup Output"
                  startId="Package\Data Flow Task\Fuzzy Lookup.Outputs
```

```
                              [Fuzzy Lookup Output]" />
                  <path
                    refId="Package\Data Flow Task.Paths[Lookup Error Output]"
                    endId="Package\Data Flow Task\Fuzzy Lookup.Inputs
                    [Fuzzy Lookup Input]"
                    name="Lookup Error Output"
                    startId="Package\Data Flow Task\Lookup.Outputs
                    [Lookup Error Output]" />
                  <path
                    refId="Package\Data Flow Task.Paths[Lookup Output]"
                    endId="Package\Data Flow Task\Multicast.Inputs
                    [Multicast Input 1]"
                    name="Lookup Output"
                    startId="Package\Data Flow Task\Lookup.Outputs
                    [Lookup Match Output]" />
              </paths>
            </pipeline>
          </DTS:ObjectData>
        </DTS:Executable>
      </DTS:Executables>
      <DTS:DesignTimeProperties><![CDATA[<?xml version="1.0"?>
<!--This CDATA section contains the layout information of the package.
The section includes information such as (x,y) coordinates, width, and
height.-->
<!--If you manually edit this section and make a mistake, you can
delete it. -->
<!--The package will still be able to load normally but the previous
layout information will be lost and the designer will automatically
re-arrange the elements on the design surface.-->
<Objects
  Version="sql11">
  <!--Each node below will contain properties that do not affect
  runtime behavior.-->
  <Package
    design-time-name="Package">
    <LayoutInfo>
      <GraphLayout
        Capacity="4" xmlns="clr-namespace:Microsoft.SqlServer.
        IntegrationServices.Designer.Model.
        Serialization;assembly=Microsoft.SqlServer.
        IntegrationServices.Graph">
        <NodeLayout
          Size="165.735849056604,43.9245283018868"
          Id="Package\Data Flow Task"
          TopLeft="5.5,5.5" />
      </GraphLayout>
    </LayoutInfo>
  </Package>
</Objects>]]></DTS:DesignTimeProperties>
    </DTS:Executable>
```

## 3.2   Web Service Task and Script Task

The following example is taken from [Haselden], with modifications.

The following package file illustrates a Web Service Task executable that draws stock quotes from a quote system and a Script Task executable that displays the result. The package file also contains a **PrecedenceConstraint** operation that describes the relationship between the two tasks. The Integration Services user interface in the designer that represents this package example is shown in the following diagram.
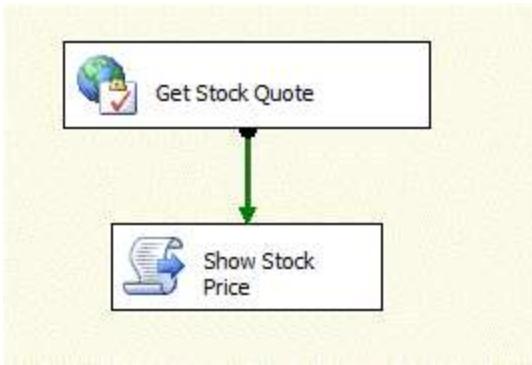
**Figure 4: Web Service Task executable and the Script Task in Integration Services user interface**

The Script Task executable in this package file is an example of how custom user code is stored in a package file. In the Script Task executable, the **ObjectData** element contains a **ScriptProject** element. Within the **ScriptProject** element is a collection of **ProjectItem** elements, each of which contains Microsoft Visual Basic source code for the script project. Also within the **ScriptProject** element is a **BinaryItem** element, which contains the compiled source code for this project, encoded as type **xs:base64binary**.

```xml
<?xml version="1.0"?>
<DTS:Executable xmlns:DTS="www.microsoft.com/SqlServer/Dts"
  DTS:refId="Package"
  DTS:CreationDate="7/16/2005 3:59:24 PM"
  DTS:CreationName="SSIS.Package.3"
  DTS:CreatorComputerName="ROMEO"
  DTS:CreatorName="Kirk Haselden"
  DTS:DTSID="{6BC04190-25CC-422A-92BE-68ACBD6F59E8}"
  DTS:EnableConfig="True"
  DTS:ExecutableType="SSIS.Package.3"
  DTS:ExecValue="0"
  DTS:ExecValueType="3"
  DTS:LastModifiedProductVersion="11.0.1153.1"
  DTS:LocaleID="1033"
  DTS:ObjectName="WebService"
  DTS:PackageType="5"
  DTS:ProtectionLevel="0"
  DTS:VersionBuild="117"
  DTS:VersionGUID="{57687733-19AA-4974-9B56-5BEB6BD40E11}">
  <DTS:Property
    DTS:Name="PackageFormatVersion">6</DTS:Property>
  <DTS:ConnectionManagers>
    <DTS:ConnectionManager
      DTS:refId="Package.ConnectionManagers[cmService]"
      DTS:CreationName="HTTP"
      DTS:DTSID="{2DA145FD-9C6E-487D-9A50-DD97C1D11763}"
      DTS:ObjectName="cmService">
      <DTS:ObjectData>
        <DTS:ConnectionManager
DTS:ConnectionString="http://www.ecubicle.net/livestockquotes.asmx?WSDL">
          <DTS:HttpConnection
DTS:ServerURL="http://www.ecubicle.net/livestockquotes.asmx?WSDL">
            <DTS:Property
              DTS:Name="ServerPassword"
              Sensitive="1"></DTS:Property>
            <DTS:Property
              DTS:Name="ProxyPassword"
              Sensitive="1"></DTS:Property>
          </DTS:HttpConnection>
        </DTS:ConnectionManager>
      </DTS:ObjectData>
```

```
            </DTS:ConnectionManager>
            <DTS:ConnectionManager
              DTS:refId="Package.ConnectionManagers[fio]"
              DTS:CreationName="FILE"
              DTS:DTSID="{C5C2675C-13D7-4DED-B499-9F45AA97C32A}"
              DTS:ObjectName="fio">
              <DTS:ObjectData>
                <DTS:ConnectionManager
                  DTS:FileUsageType="1"
                  DTS:ConnectionString="fio" />
              </DTS:ObjectData>
            </DTS:ConnectionManager>
          </DTS:ConnectionManagers>
          <DTS:Configurations>
            <DTS:Configuration
              DTS:ConfigurationString="SSISSAMPLESROOT"
              DTS:ConfigurationType="2"
              DTS:ConfigurationVariable="\Package.Variables[SAMPLES::ROOTWD].
              Properties[Value]"
              DTS:CreationName=""
              DTS:DTSID="{90405045-3A91-43C2-B759-6C183C0E81A6}"
              DTS:ObjectName="Configure The Root Folder" />
            <DTS:Configuration
              DTS:ConfigurationString="SSISSAMPLESERVER"
              DTS:ConfigurationType="2"
              DTS:ConfigurationVariable="\Package.Variables[SAMPLES::SERVERNAME].
              Properties[Value]"
              DTS:CreationName=""
              DTS:DTSID="{A9F64F24-15C2-4198-A308-F9124C48468D}"
              DTS:ObjectName="Configure The Sample Server Name" />
          </DTS:Configurations>
          <DTS:Variables>
            <DTS:Variable
              DTS:CreationName=""
              DTS:DTSID="{E05C4384-FA46-4EDB-9E41-E0CDD6DD4DC6}"
              DTS:EvaluateAsExpression="True"
              DTS:Expression="@ROOTWD + &quot;\\&quot; + @SAMPLESUBFOLDER"
              DTS:IncludeInDebugDump="2345"
              DTS:Namespace="SAMPLES"
              DTS:ObjectName="CURRENTWD">
              <DTS:VariableValue
                DTS:DataType="8">C:\USERS\V-TOMCON.REDMOND\MY DOCUMENTS\SSIS\
                BOOKSAMPLES2005\S08-StockTasks</DTS:VariableValue>
            </DTS:Variable>
            <DTS:Variable
              DTS:CreationName=""
              DTS:DTSID="{3BD365BA-FE06-4B5A-8100-69F27C71B74E}"
              DTS:IncludeInDebugDump="2345"
              DTS:Namespace="SAMPLES"
              DTS:ObjectName="ROOTWD">
              <DTS:VariableValue
                DTS:DataType="8">C:\USERS\V-TOMCON.REDMOND\MY DOCUMENTS\SSIS\
                BOOKSAMPLES2005</DTS:VariableValue>
            </DTS:Variable>
            <DTS:Variable
              DTS:CreationName=""
              DTS:DTSID="{6E7FB8C5-7F98-454E-87E0-D9BF0D0E06AD}"
              DTS:IncludeInDebugDump="2345"
              DTS:Namespace="SAMPLES"
              DTS:ObjectName="SAMPLESUBFOLDER">
              <DTS:VariableValue
                DTS:DataType="8">S08-StockTasks</DTS:VariableValue>
            </DTS:Variable>
            <DTS:Variable
              DTS:CreationName=""
              DTS:DTSID="{5F9998D0-186D-4C44-A949-66C8477A3CE7}"
              DTS:IncludeInDebugDump="2345"
              DTS:Namespace="SAMPLES"
              DTS:ObjectName="SERVERNAME">
              <DTS:VariableValue
```

```
              DTS:DataType="8">.</DTS:VariableValue>
        </DTS:Variable>
        <DTS:Variable
          DTS:CreationName=""
          DTS:DTSID="{E084171E-93C0-43BC-BC6C-83833B78109E}"
          DTS:IncludeInDebugDump="2345"
          DTS:Namespace="SAMPLES"
          DTS:ObjectName="STOCKQUOTE">
          <DTS:VariableValue
            DTS:DataType="8"></DTS:VariableValue>
        </DTS:Variable>
        <DTS:Variable
          DTS:CreationName=""
          DTS:DTSID="{58589E35-3D84-4D07-8A32-A75E6591D7F0}"
          DTS:IncludeInDebugDump="2345"
          DTS:Namespace="SAMPLES"
          DTS:ObjectName="TICKERSYMBOL">
          <DTS:VariableValue
            DTS:DataType="8">MSFT</DTS:VariableValue>
        </DTS:Variable>
      </DTS:Variables>
      <DTS:Executables>
        <DTS:Executable
          DTS:refId="Package\Get Stock Quote"
          DTS:CreationName="Microsoft.SqlServer.Dts.Tasks.WebServiceTask.
          WebServiceTask, Microsoft.SqlServer.WebServiceTask,
          Version=11.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91"
          DTS:Description="Web Service Task"
          DTS:DTSID="{6CA7FC7C-AB58-4C39-BD00-3F22B51AAB92}"
          DTS:ExecutableType="Microsoft.SqlServer.Dts.Tasks.WebServiceTask.
          WebServiceTask, Microsoft.SqlServer.WebServiceTask, Version=11.0.0.0,
          Culture=neutral, PublicKeyToken=89845dcd8080cc91"
          DTS:ExecValue="0"
          DTS:ExecValueType="3"
          DTS:LocaleID="-1"
          DTS:ObjectName="Get Stock Quote"
          DTS:TaskContact="Web Service Task; Microsoft Corporation; Microsoft
          SQL Server v9; © 2004 Microsoft Corporation; All Rights Reserved;
          http://www.microsoft.com/sql/support/default.asp;1"
          DTS:ThreadHint="0">
          <DTS:Variables />
          <DTS:PropertyExpression
            DTS:Name="WsdlFile">@[SAMPLES::ROOTWD] + "\\SampleData\\Service.wsdl"
          </DTS:PropertyExpression>
          <DTS:ObjectData>
            <WSTask:WebServiceTaskData
              WSTask:ConnectionName="{2DA145FD-9C6E-487D-9A50-DD97C1D11763}"
              WSTask:Service="LiveStockQuotes"
              WSTask:WSDLFile="C:\USERS\V-TOMCON.REDMOND\MY DOCUMENTS\SSIS\
              BOOKSAMPLES2005\SampleData\Service.wsdl"
              WSTask:OverwriteWSDLFile="True"
              WSTask:OutputLocation="fio" xmlns:WSTask="www.microsoft.com/
              sqlserver/dts/tasks/webservicetask">
              <WSTask:MethodInfo
                WSTask:MethodName="liveMarketAsString"
                WSTask:MessageName="LiveMarketAsString">
                <WSTask:Documentation>&amp;lt;h3&amp;gt;This method can be used
                if a consumer is unable to consume liveMarket method. The
                returned data is XML string.&amp;lt;/h3&amp;gt;
              </WSTask:Documentation>
                <WSTask:ParamInfo
                  WSTask:Name="stockSymbol"
                  WSTask:Datatype="string"
                  WSTask:ParamType="Primitive"
                  WSTask:SeqNumber="0">
                  <WSTask:PrimitiveValue></WSTask:PrimitiveValue>
                </WSTask:ParamInfo>
              </WSTask:MethodInfo>
            </WSTask:WebServiceTaskData>
          </DTS:ObjectData>
```

```
    </DTS:Executable>
    <DTS:Executable
      DTS:refId="Package\Show Stock Price"
      DTS:CreationName="Microsoft.SqlServer.Dts.Tasks.ScriptTask.ScriptTask,
      Microsoft.SqlServer.ScriptTask, Version=11.0.0.0, Culture=neutral,
      PublicKeyToken=89845dcd8080cc91"
      DTS:Description="Script Task"
      DTS:DTSID="{325ba39d-e3ab-46cc-8f27-c28e9a5f9f8f}"
      DTS:ExecutableType="Microsoft.SqlServer.Dts.Tasks.ScriptTask.
      ScriptTask, Microsoft.SqlServer.ScriptTask, Version=11.0.0.0,
      Culture=neutral, PublicKeyToken=89845dcd8080cc91"
      DTS:ExecValue="0"
      DTS:ExecValueType="3"
      DTS:LocaleID="-1"
      DTS:ObjectName="Show Stock Price"
      DTS:ThreadHint="0">
      <DTS:Variables />
      <DTS:ObjectData>
        <ScriptProject
          Name="ScriptTask_053aaa1de6d84acab80050be0c42b6f2"
          VSTAMajorVersion="2"
          VSTAMinorVersion="2"
          Language="VisualBasic"
          ReadOnlyVariables="SAMPLES::STOCKQUOTE,SAMPLES::TICKERSYMBOL">
          <ProjectItem
            Name="\my project\assemblyinfo.vb"><![CDATA[Imports System
Imports System.Reflection
Imports System.Runtime.InteropServices
' General Information about an assembly is controlled through the following
' set of attributes. Change these attribute values to modify the information
' associated with an assembly.
' Review the values of the assembly attributes
<Assembly: AssemblyTitle("ScriptTask_053aaa1de6d84acab80050be0c42b6f2.vbproj")>
<Assembly: AssemblyDescription("")>
<Assembly: AssemblyCompany("MSIT")>
<Assembly: AssemblyProduct("ScriptTask_053aaa1de6d84acab80050be0c42b6f2.vbproj")>
<Assembly: AssemblyCopyright("Copyright @ MSIT 2008")>
<Assembly: AssemblyTrademark("")>
<Assembly: CLSCompliant(True)>
<Assembly: ComVisible(False)>
'The following GUID is for the ID of the typelib if this project is exposed to COM
<Assembly: Guid("e54931f7-e817-4314-a409-3cdc9702928a")>
' Version information for an assembly consists of the following four values:
'
'      Major Version
'      Minor Version
'      Build Number
'      Revision
'
' You can specify all the values or you can default the Build and Revision Numbers
' by using the '*' as shown below:
' <Assembly: AssemblyVersion("1.0.*")>
<Assembly: AssemblyVersion("1.0.0.0")>
<Assembly: AssemblyFileVersion("1.0.0.0")> ]]></ProjectItem>
          <ProjectItem
            Name="\my project\settings.designer.vb"><![CDATA['--------------------------
--------------------------------------------
' <autogenerated>
'     This code was generated by a tool.
'
'     Changes to this file may cause incorrect behavior and will be lost if
'     the code is regenerated.
' </autogenerated>
'------------------------------------------------------------------------
Option Strict Off
Option Explicit On
Partial Friend NotInheritable Class MySettings
    Inherits System.Configuration.ApplicationSettingsBase
    Private Shared m_Value As MySettings
    Private Shared m_SyncObject As Object = New Object
```

```
            <System.Diagnostics.DebuggerNonUserCode()> _
        Public Shared ReadOnly Property Value() As MySettings
            Get
                If (MySettings.m_Value Is Nothing) Then
                    System.Threading.Monitor.Enter(MySettings.m_SyncObject)
                    If (MySettings.m_Value Is Nothing) Then
                        Try
                            MySettings.m_Value = New MySettings
                        Finally
                            System.Threading.Monitor.Exit(MySettings.m_SyncObject)
                        End Try
                    End If
                End If
                Return MySettings.m_Value
            End Get
        End Property
    End Class]]></ProjectItem>
            <ProjectItem
            Name="\scripttask_053aaa1de6d84acab80050be0c42b6f2.vbproj">
            <![CDATA[<Project ToolsVersion="3.5" DefaultTargets="Build"
            xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
    <!-- This section defines project-level properties.
        Configuration - Specifies whether the default
        configuration is Release or Debug.
        Platform - Specifies what CPU the output of this project can
        run on.
        OutputType - Must be "Library" for VSTA.
        NoStandardLibraries - Set to "false" for VSTA.
        RootNamespace - In C#, this specifies the namespace given to
        new files.
                    In Visual Basic, all objects are wrapped in
                    this namespace at runtime.
        AssemblyName - Name of the output assembly.
    -->
    <PropertyGroup>
      <ProjectTypeGuids>{A860303F-1F3F-4691-B57E-529FC101A107};
      {F184B08F-C81C-45F6-A57F-5ABD9991F28F}</ProjectTypeGuids>
      <Configuration Condition=" '$(Configuration)' == '' ">Debug
      </Configuration>
      <Platform Condition=" '$(Platform)' == '' ">AnyCPU</Platform>
      <OutputType>Library</OutputType>
      <RootNamespace>ScriptTask_053aaa1de6d84acab80050be0c42b6f2.vbproj
      </RootNamespace>
      <AssemblyName>ScriptTask_053aaa1de6d84acab80050be0c42b6f2.vbproj
      </AssemblyName>
      <StartupObject>
      </StartupObject>
      <OptionExplicit>On</OptionExplicit>
      <OptionCompare>Binary</OptionCompare>
      <OptionStrict>Off</OptionStrict>
      <OptionInfer>On</OptionInfer>
      <ProjectGuid>{EB2B3188-9A8D-427A-988F-48D11FE940F1}</ProjectGuid>
    </PropertyGroup>
    <!-- This section defines properties that are set when the "Debug"
    configuration is
        selected.
        DebugSymbols - If true, create symbols (.pdb). If false, do not
        create symbols.
        Optimize - If true, optimize the build output. If false, do
        not optimize.
        OutputPath - Output path of the project relative to the
        project file.
        EnableUnmanagedDebugging - If true, starting the debugger
        will attach both managed and unmanaged debuggers.
        DefineConstants - Constants defined for the preprocessor.
        Warning Level - Warning level for the compiler.
    -->
    <PropertyGroup Condition=" '$(Configuration)' == 'Debug' ">
      <DefineConstants>
      </DefineConstants>
```

```
        <DefineDebug>true</DefineDebug>
        <DefineTrace>true</DefineTrace>
        <DebugSymbols>true</DebugSymbols>
        <Optimize>false</Optimize>
        <OutputPath>bin\</OutputPath>
        <RegisterForComInterop>false</RegisterForComInterop>
        <RemoveIntegerChecks>false</RemoveIntegerChecks>
        <NoWarn>42016,42017,42018,42019,42032</NoWarn>
    </PropertyGroup>
    <!-- This section defines properties that are set when the "Release"
    configuration is
        selected.
        DebugSymbols - If true, create symbols (.pdb). If false,
        do not create symbols.
        Optimize - If true, optimize the build output. If false,
        do not optimize.
        OutputPath - Output path of the project relative to the
        project file.
        EnableUnmanagedDebugging - If true, starting the debugger will
        attach both managed and unmanaged debuggers.
        DefineConstants - Constants defined for the preprocessor.
        Warning Level - Warning level for the compiler.
    -->
    <PropertyGroup Condition=" '$(Configuration)' == 'Release' ">
        <DefineConstants>
        </DefineConstants>
        <DefineDebug>false</DefineDebug>
        <DefineTrace>true</DefineTrace>
        <DebugSymbols>false</DebugSymbols>
        <Optimize>true</Optimize>
        <OutputPath>bin\</OutputPath>
        <RegisterForComInterop>false</RegisterForComInterop>
        <RemoveIntegerChecks>false</RemoveIntegerChecks>
        <NoWarn>42016,42017,42018,42019,42032</NoWarn>
    </PropertyGroup>
    <!-- This section enables pre- and post-build steps. However,
        it is recommended that MSBuild tasks be used instead of these
        properties.
    -->
    <PropertyGroup>
        <PreBuildEvent>
        </PreBuildEvent>
        <PostBuildEvent>
        </PostBuildEvent>
    </PropertyGroup>
    <!-- This sections specifies references for the project. -->
    <ItemGroup>
        <Reference Include="Microsoft.SqlServer.ManagedDTS,
        Version=11.0.0.0, Culture=neutral,
        PublicKeyToken=89845dcd8080cc91" />
        <Reference Include="Microsoft.SqlServer.ScriptTask,
        Version=11.0.0.0, Culture=neutral,
        PublicKeyToken=89845dcd8080cc91" />
        <Reference Include="System" />
        <Reference Include="System.AddIn,
        Version=3.5.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089,
        processorArchitecture=MSIL" />
        <Reference Include="System.Data" />
        <Reference Include="System.Windows.Forms" />
        <Reference Include="System.Xml" />
    </ItemGroup>
    <!-- Visual Basic supports Importing namespaces
    (equivalent to using statements in C#).-->
    <ItemGroup>
        <Import Include="Microsoft.VisualBasic" />
        <Import Include="System" />
        <Import Include="System.Collections" />
        <Import Include="System.Data" />
        <Import Include="System.Diagnostics" />
        <Import Include="System.Windows.Forms" />
```

```
    </ItemGroup>
    <!-- This section defines the user source files that are part of the
         project.
         Compile - Specifies a source file to compile.
         EmbeddedResource - Specifies a .resx file for embedded resources.
         None - Specifies a file that is not to be passed to the compiler
         (for instance,
                a text file or XML file).
         AppDesigner - Specifies the directory where the application
         properties files can
                       be found.
    -->
    <ItemGroup>
      <AppDesigner Include="My Project\" />
      <Compile Include="My Project\AssemblyInfo.vb">
        <SubType>Code</SubType>
      </Compile>
      <EmbeddedResource Include="My Project\Resources.resx">
        <Generator>VbMyResourcesResXFileCodeGenerator</Generator>
        <LastGenOutput>Resources.Designer.vb</LastGenOutput>
        <CustomToolNamespace>My.Resources</CustomToolNamespace>
      </EmbeddedResource>
      <Compile Include="My Project\Resources.Designer.vb">
        <AutoGen>True</AutoGen>
        <DesignTime>True</DesignTime>
        <DependentUpon>Resources.resx</DependentUpon>
        <SubType>Code</SubType>
      </Compile>
      <None Include="My Project\Settings.settings">
        <Generator>SettingsSingleFileGenerator</Generator>
        <LastGenOutput>Settings.Designer.vb</LastGenOutput>
      </None>
      <Compile Include="My Project\Settings.Designer.vb">
        <AutoGen>True</AutoGen>
        <DependentUpon>Settings.settings</DependentUpon>
        <SubType>Code</SubType>
      </Compile>
      <Compile Include="ScriptMain.designer.vb">
        <DependentUpon>ScriptMain.Designer.xml</DependentUpon>
      </Compile>
      <Compile Include="ScriptMain.vb">
        <SubType>Code</SubType>
      </Compile>
      <None Include="ScriptMain.Designer.xml">
        <DependentUpon>ScriptMain.vb</DependentUpon>
      </None>
      <!-- Include the default configuration information and metadata
      files for the add-in.
           These files are copied to the build output directory when
           the project is
           built, and the path to the configuration file is passed to
           add-in on the command
           line when debugging.
      -->
    </ItemGroup>
    <!-- Include the build rules for a VB project.-->
    <Import Project="$(MSBuildBinPath)\Microsoft.VisualBasic.targets" />
    <!-- This section defines VSTA properties that describe the host-
    changeable project properties. -->
    <ProjectExtensions>
      <VisualStudio>
        <FlavorProperties GUID="{A860303F-1F3F-4691-B57E-529FC101A107}">
          <ProjectProperties HostName="VSTAHostName" HostPackage=
          "{C1B21C64-9E6F-4923-A89D-9F958503C1CE}" ApplicationType="usd"
          Language="vb" TemplatesPath="" DebugInfoExeName=
          "#HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\VisualStudio\9.0\
          InstallDir#\devenv.exe" />
          <Host Name="SSIS_ScriptTask">
            <HostItem Name="ScriptMain" Code="ScriptMain.vb"
            CanonicalName="ScriptMain" DisplayName="ScriptMain"
```

```
                    Blueprint="ScriptMain.Designer.xml" GeneratedCode=
                    "ScriptMain.designer.vb" />
                </Host>
                <ProjectClient>
                    <HostIdentifier>SSIS_ST110</HostIdentifier>
                </ProjectClient>
            </FlavorProperties>
        </VisualStudio>
    </ProjectExtensions>
</Project>]]></ProjectItem>
            <ProjectItem
              Name="\my project\settings.settings">
              <![CDATA[<?xml version='1.0' encoding='iso-8859-1'?>
<SettingsFile xmlns="uri:settings" CurrentProfile="(Default)"
GeneratedClassNamespace="$safeprojectname" GeneratedClassName="MySettings">
  <Profiles>
    <Profile Name="(Default)" />
  </Profiles>
  <Settings />
</SettingsFile>]]></ProjectItem>
            <ProjectItem
              Name="\my project\resources.designer.vb"><![CDATA['------------------------
-------------------------------------------
' <autogenerated>
'     This code was generated by a tool.
'
'     Changes to this file may cause incorrect behavior and will be lost if
'     the code is regenerated.
' </autogenerated>
'------------------------------------------------------------------------
Option Strict Off
Option Explicit On
Namespace My.Resources
    '''<summary>
    '''    A strongly-typed resource class, for looking up localized strings,
    etc.
    '''</summary>
    'This class was auto-generated by the Strongly Typed Resource Builder
    'class via a tool like ResGen or Visual Studio.NET.
    'To add or remove a member, edit your .ResX file then rerun ResGen
    'with the /str option, or rebuild your VS project.
    Class MyResources
        Private Shared _resMgr As System.Resources.ResourceManager
        Private Shared _resCulture As System.Globalization.CultureInfo
        Friend Sub New()
            MyBase.New
        End Sub
        '''<summary>
        '''    Returns the cached ResourceManager instance used by this class.
        '''</summary>
        <System.ComponentModel.EditorBrowsableAttribute(System.
        ComponentModel.EditorBrowsableState.Advanced)> _
        Public Shared ReadOnly Property ResourceManager() As System.
        Resources.ResourceManager
            Get
                If (_resMgr Is Nothing) Then
                    Dim temp As System.Resources.ResourceManager =
                    New System.Resources.ResourceManager
                    ("My.Resources.MyResources",
                    GetType(MyResources).Assembly)
                    _resMgr = temp
                End If
                Return _resMgr
            End Get
        End Property
        '''<summary>
        '''    Overrides the current thread's CurrentUICulture
        property for all
        '''    resource lookups using this strongly typed resource class.
        '''</summary>
```

```
        <System.ComponentModel.EditorBrowsableAttribute
        (System.ComponentModel.EditorBrowsableState.Advanced)> _
        Public Shared Property Culture() As System.Globalization.CultureInfo
            Get
                Return _resCulture
            End Get
            Set
                _resCulture = value
            End Set
        End Property
    End Class
End Namespace]]></ProjectItem>
        <ProjectItem
          Name="\scriptmain.designer.vb">
          <![CDATA[<System.CLSCompliantAttribute(False)> _
Partial Public Class ScriptMain
Inherits Microsoft.SqlServer.Dts.Tasks.ScriptTask.
  VSTARTScriptObjectModelBase
End Class]]></ProjectItem>
        <ProjectItem
          Name="\my project\resources.resx">
          <![CDATA[<?xml version="1.0" encoding="utf-8"?>
<root>
  <!--
    Microsoft ResX Schema
    Version 2.0
    The primary goals of this format is to allow a simple XML format
    that is mostly human readable. The generation and parsing of the
    various data types are done through the TypeConverter classes
    associated with the data types.
    Example:
    ... ado.net/XML headers & schema ...
    <resheader name="resmimetype">text/microsoft-resx</resheader>
    <resheader name="version">2.0</resheader>
    <resheader name="reader">System.Resources.ResXResourceReader,
    System.Windows.Forms, ...</resheader>
    <resheader name="writer">System.Resources.ResXResourceWriter,
    System.Windows.Forms, ...</resheader>
    <data name="Name1"><value>this is my long string</value><comment>
    this is a comment</comment></data>
    <data name="Color1" type="System.Drawing.Color, System.Drawing">Blue
    </data>
    <data name="Bitmap1" mimetype="application/x-microsoft.net.object.
    binary.base64">
        <value>[base64 mime encoded serialized .NET Framework object]
        </value>
    </data>
    <data name="Icon1" type="System.Drawing.Icon, System.Drawing"
    mimetype="application/x-microsoft.net.object.bytearray.base64">
        <value>[base64 mime encoded string representing a byte array
        form of the .NET Framework object]</value>
        <comment>This is a comment</comment>
    </data>
    There are any number of "resheader" rows that contain simple
    name/value pairs.
    Each data row contains a name, and value. The row also contains a
    type or mimetype. Type corresponds to a .NET class that support
    text/value conversion through the TypeConverter architecture.
    Classes that don't support this are serialized and stored with the
    mimetype set.
    The mimetype is used for serialized objects, and tells the
    ResXResourceReader how to depersist the object. This is currently not
    extensible. For a given mimetype the value must be set accordingly:
    Note - application/x-microsoft.net.object.binary.base64 is the format
    that the ResXResourceWriter will generate, however the reader can
    read any of the formats listed below.
    mimetype: application/x-microsoft.net.object.binary.base64
    value   : The object must be serialized with
            : System.Serialization.Formatters.Binary.BinaryFormatter
            : and then encoded with base64 encoding.
```

```
            mimetype: application/x-microsoft.net.object.soap.base64
            value   : The object must be serialized with
                    : System.Runtime.Serialization.Formatters.Soap.SoapFormatter
                    : and then encoded with base64 encoding.
            mimetype: application/x-microsoft.net.object.bytearray.base64
            value   : The object must be serialized into a byte array
                    : using a System.ComponentModel.TypeConverter
                    : and then encoded with base64 encoding.
        -->
    <xsd:schema id="root" xmlns="" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
        <xsd:element name="root" msdata:IsDataSet="true">
            <xsd:complexType>
                <xsd:choice maxOccurs="unbounded">
                    <xsd:element name="metadata">
                        <xsd:complexType>
                            <xsd:sequence>
                                <xsd:element name="value" type="xsd:string" minOccurs="0" />
                            </xsd:sequence>
                            <xsd:attribute name="name" type="xsd:string" />
                            <xsd:attribute name="type" type="xsd:string" />
                            <xsd:attribute name="mimetype" type="xsd:string" />
                        </xsd:complexType>
                    </xsd:element>
                    <xsd:element name="assembly">
                        <xsd:complexType>
                            <xsd:attribute name="alias" type="xsd:string" />
                            <xsd:attribute name="name" type="xsd:string" />
                        </xsd:complexType>
                    </xsd:element>
                    <xsd:element name="data">
                        <xsd:complexType>
                            <xsd:sequence>
                                <xsd:element name="value" type="xsd:string"
                                minOccurs="0" msdata:Ordinal="1" />
                                <xsd:element name="comment" type="xsd:string"
                                minOccurs="0" msdata:Ordinal="2" />
                            </xsd:sequence>
                            <xsd:attribute name="name" type="xsd:string"
                            msdata:Ordinal="1" />
                            <xsd:attribute name="type" type="xsd:string"
                            msdata:Ordinal="3" />
                            <xsd:attribute name="mimetype" type="xsd:string"
                            msdata:Ordinal="4" />
                        </xsd:complexType>
                    </xsd:element>
                    <xsd:element name="resheader">
                        <xsd:complexType>
                            <xsd:sequence>
                                <xsd:element name="value" type="xsd:string"
                                minOccurs="0" msdata:Ordinal="1" />
                            </xsd:sequence>
                            <xsd:attribute name="name" type="xsd:string" use="required" />
                        </xsd:complexType>
                    </xsd:element>
                </xsd:choice>
            </xsd:complexType>
        </xsd:element>
    </xsd:schema>
    <resheader name="resmimetype">
        <value>text/microsoft-resx</value>
    </resheader>
    <resheader name="version">
        <value>2.0</value>
    </resheader>
    <resheader name="reader">
        <value>System.Resources.ResXResourceReader, System.Windows.Forms,
        Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>
    </resheader>
    <resheader name="writer">
```

```
        <value>System.Resources.ResXResourceWriter, System.Windows.Forms,
      Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>
    </resheader>
</root>]]></ProjectItem>
            <ProjectItem
              Name="\scriptmain.vb"><![CDATA[Imports System
Imports System.Data
Imports System.Math
Imports Microsoft.SqlServer.Dts.Runtime
Imports System.Xml
<System.AddIn.AddIn("ScriptMain", Version:="1.0", Publisher:="",
Description:="")> _
Partial Class ScriptMain
Private Sub ScriptMain_Startup(ByVal sender As Object, ByVal e As
  System.EventArgs)
End Sub
Private Sub ScriptMain_Shutdown(ByVal sender As Object, ByVal e As
  System.EventArgs)
Try
' Unlock variables from the read-only and read-write variable
      collection properties
If (Dts.Variables.Count <> 0) Then
Dts.Variables.Unlock()
End If
Catch ex As Exception
End Try
End Sub
Enum ScriptResults
Success = Microsoft.SqlServer.Dts.Runtime.DTSExecResult.Success
Failure = Microsoft.SqlServer.Dts.Runtime.DTSExecResult.Failure
End Enum
    Public Sub Main()
        Dim last As String = New String("")
        Dim xmlDoc As XmlDocument = New XmlDocument()
        xmlDoc.LoadXml(Dts.Variables("SAMPLES::STOCKQUOTE").Value.ToString())
        If Not IsDBNull(xmlDoc) Then
            Dim list As XmlNodeList
            list = xmlDoc.DocumentElement.GetElementsByTagName("last_trade")
            If list.Count > 0 Then
                last = list.Item(0).InnerText
                MsgBox("The last price for the stock " + Dts.Variables
                ("SAMPLES::TICKERSYMBOL").Value.ToString() + ": is " +
                last.ToString(), MsgBoxStyle.OkOnly,
                "Webservices Task Sample")
                Dts.TaskResult = ScriptResults.Success
            End If
        End If
    End Sub
End Class]]></ProjectItem>
          <ProjectItem
            Name="\scriptmain.designer.xml">
            <![CDATA[<hostitem:hostItem hostitem:identifier="ScriptMain"
            hostitem:namespace="" hostitem:primaryCookie="ScriptMain"
hostitem:baseType="Microsoft.SqlServer.Dts.Tasks.ScriptTask.VSTAProxy.
            VSTARTScriptObjectModelBaseEntryPoint"
            hostitem:interfaceType=
            "Microsoft.VisualStudio.Tools.Applications.
            Runtime.IEntryPoint" hostitem:displayName="ScriptMain"
            hostitem:generateCode="true" hostitem:startupIndex="0"
            xmlns:hostitem=
"http://schemas.microsoft.com/2006/VisualStudio/Tools/Applications/
HostItem.xsd" />]]></ProjectItem>
          <BinaryItem
            Name="\bin\scripttask_053aaa1de6d84acab80050be0c42b6f2.vbproj.dll">
TVqQAAMAAAAEAAAA//8AALgAAAAAAAAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAgAAAAA4fug4AtAnNIbgBTM0hVGhpcyBwcm9ncmFtIGNhbm5vdCBiZSBydW4gaW4gRE9TIG1v
ZGUuDQ0KJAAAAAAAAABQRQAATAEDAKSIBk0AAAAAAAAAAAOAAAiELAQgAAB4AAAAMAAAAAAAzj0A
AAAgAAAAQAAAABAAAAgAAAAgAABAAAAAAAAAAEAAAAAAAACAAAAAgAAAAAAIAQIUAABBAA
ABAAAAAAEAAAEAAAAAABAAAAAAAAAAAAHg9AABTAAAAEAAAIgJAAAAAAAAAAAAAAAAAAAA
AAAAGAAAwAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

AAAAIAAACAAAAAAAAAAAAAAAACCAAAEgAAAAAAAAAAAAAAC50ZXh0AAAA1B0AAAAgAAAAHgAAAIA
AAAAAAAAAAAAAAAACAAAGAucnNyYwAAAIgJAAAAQAAAAoAAAAgAAAAAAAAAAAAAAAAAAAABAAAAABA
LnJlbG9jAAAAMAAAAAAGAAAAACAAAAKgAAAAAAAAAAAAAAAAAAAQgAAAAAAAAAAAAAAAAAAACw
PQAAAAAAAEgAAAACAAUAgCQAAPgYAAABAAAAAAAAAFAgAAC4AAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAALQAAADOyu++AQAAAJEAAABsU3lzdGVtLlJlc291cmNlcy5S
ZXNvdXJjZVJlYWRlciwgbXNjb3JsaWIsIFZlcnNpb249Mi4wLjAuMCwgQ3VsdHVyZT1uZXV0cmFs
LCBQdWJsaWNLZXlUb2tlbj1iNzdhNWM1NjE5MzRlMDg5I1N5c3RlbS5SZXNvdXJjZXMuUnVudGlt
ZVJlc291cmNlU2V0AgAAAAAAAAAAAAAUEFUEUFEULQAAAeAigBAAAKh4CKAQAAAoqpnMGAAAK
gAEAAARzBwAACoACAAAEcwgAAAAqAAwAABHMJAAAKgAQAAAQqAAAATMAEACwAAAAEAABF+AQAABG8K
AAAKKgATMAEACwAAAAIAABF+AgAABG8LAAAKKgATMAEACwAAAAMAABF+AwAABG8MAAAKKgATMAEA
CwAAAQAABF+BAAABG8NAAAKKgATMAIADQAAAAUAABECAygRAAAKKBIAAAoqAAAAAEzABAACAAAAAG
AAARAigTAAAKKgATMAEACwAAAcAABHQBQAAAigUAAAKKgATMAEABwAAAAgAABECKBUAAAoqABMw
AQAQAAAACQAAEQKMAQAAGy0GKAEAACsqAioTMAIAEAAAAAAoAABEDEgD+FQIAABsGgQIAABsqHgIo
FwAACioTMAIAKAAAAAsAABECexkAAApvGgAACgsHjAMAABstEigCAAArCwJ7GQAACgdvGwAACgcq
SgIoFwAACgJzHAAAACn0ZAAAKKgAeAigXAAAKKhHMwAgAtAAAADAAAEX4GAAAELSByAQAAcNAHAAAC
KBQAAApvHgAACnMfAAAKCweABgAABH4GAAAEKgAAAMwAQAGAAAADQAAEX4HAAAEKgAAHgKABwAA
BCpCcxcAAAooEQAACoAJAAAEKgAAB4CKAAAAoqGzABAD8AAAAOAAARfggAAAQtMn4JAAAAEKBEA
AAooIQAACn4IAAAAELRxzFgAABoAIAAAE3hB+CQAABCgRAAAKKCIAAAarcfggAAAQqAAEMAAACAB0A
DCkAEAAAAAEeAigkAAAKKgYqAAAbMAIAANAAAAAA8AABECbyUAAApvJgAACm8nAAAAKFi4QAm8lAAAK
byYAAApvKAAACt4OJSgpAAAKCigqAAAK3gAqAQwAAAAAAlJQAOHAAAARMwBAC3AAAAEAAAEXIz
AABwKCsAAAApzLAAACgpzLQAACgsHAm8lAAAKbyYAAApyNQAAcG8uAAAKby8AAApvFQAACg8wAAAK
BygxAAAKLXMHbzIAAApyXQAAcG8zAAAKDAhvNAAAChYxWQgWbzUAAApvNgAACgpycwAAAAAAJvJQAA
Cm8mAAAKcq8AAHBvLgAACm8vAAAKbxUAAApy2wAACAZvNwAACig4AAAKFnLnLnAABwKDkAAAomAm8l
AAAKFm86AAAKKgBCU0pCAQABAAAAAAAMAAAAdjIuMC41MDcyNwAAAAFAGwAACsCAAAI34AABgJ
AADMCgAAI1N0cmluZ3M3MAAAABgAAjVVMA/BQAABAAAAjR1VJRAAAAwwAVAADsAwAAAI0Js
b2IAAAAAAAAAgAAAVcdogkJDwAAAPoBMwAAAABAAAAAAWAAAAoAAAMAAAAAGwAAAAgAAABGAAAA
AgAAAAADUAAAQAAAABAAAAJAAAACQAAAEAAAAHAAAAQAAAAMAAAAAgAAAAAkwoB
AAAAAAKAHgBTgEKAKwBjgEGALWBtQEKABQCTgEGAJICtQEKABQD6gIGAEoDOQMGAHcDYgMOAN4D
yQMAF0ENAQGAHkEtQEGALgEtQEOAOsE1QQOAAQF1QQOADEFYAUFwUGAGAFsGnAYGAOEGzwYGAAgH9wYGABsHSAUGADgH
tQESAEIHNAQWAJsHewcWALMHewcKAN4HcwUUaABcIDAgaACMIDAgKAC8IcwUGAE4ItQEWAFUIewcK
AG8IEwAaAAIQIDAgaALgIDAgKANoIEwAKAKOYIEwAKAPMIQEeEDgJKwkgJKwk4J/AUGAK4J/J
/AUGAMwJzwYGAOk0JnAYGAB1KzwYGAGAEYKKzwYGAF8KzwYGAM8wAAAK
AAAAAQABAAAAAAAApADcABQABAAEAAAAAAG0ANwAJAAEAGAAAAAAAGAAAAAQAeaRAA3AAyQADAAUCAAA
DQAFAAgABQEAAAJAAAANAAUADwAAAAAqwC3AA0BgARAAAEAD3AAIBJQAIABUAAQAAADUBAgEp
AAoAGAACAQAAQAEAADEACgAcADE1wEgAEDAAAItADEAiG6ADEARwJHACAwOQABEAWgOdABEA
gwOhABEA9gO/ABEA/gPDAAAyGvQTgAFaAxQTjAFaAzQTjAAghAAAAAAYYiAETAAEAAECEAAAABhiI
ARMAAAAQAYIQAAAAaAARGMMBFwABAEQQhAAAAABMIygEbAAEAXCEAAAAAEwjwASgAAQB0IQAAAATCBkC
NAQBAAIwhAAAAABMINwJCAAEApCEAAAARgJ9AmMAQADAIQAAAAEBGAoYCaAACANQhAAAAAIMAlwJs
AAIA7CEAAAARgKfAnEAAgAAAARKgCeAACBwiAAAAAAAxgKAAMA0CIAAAAAAGMMBFwABAEQhAAAAAQhjwvAAASgAAQB0IQAAAAFgivA68ABADoAgAAAAARGMMBFwBAFAPwiAAAAAAYYiETAAUA
BCMAAAAAFggLBMYABQBgIwAAAGGGBEwAFAGGjAAAAAEAgwTZAAUAbCMAAAAAQCfBNkABwC8
IwAAAAGALMEEwAJAAAAAQCEAgAAAQC9AgAAQC9AgAAQC7AwAAAQCWBAAAAgCdBAAAAQCWBAAA
AgCdBAkAiAETAGkAiAHxAHkAiAEAAREAiAETAIEAiAETACQAiAETACwAiAETADQAiAETADwAiAET
ACQA2gKLACwA2gKLADQA2gKLADwA2gKLAIkAiAETAJEAiAETAJkAiAFOAaEAKwaZARkAfQJjABkA
hgJoCkATAamARkAnwJxkAiAETALALGAUQAIwIwAAAAAiAETANAEAEd8AtEAfgd8Atk0AiAETAFEAiAET
AMEAiAFQAikA6gZbAjkAiAFgAkkAiAETANEAEAEd8AtEAFEAiAETAFEAVAeGAukA
pQeLAvkAzQdoAPEA1wcTAAEB6geQAGEB+gcXABkEBzQdoABEBwAjEAkEBnwJxACEB0wjLAkkB
/wjTAukABgneAmEBiAFQAmkBiAFOAXEBiAHeAnkBiAETAPEAXginAikBCwSu
AgkBZwOoATEBewiyAgkBjwi3AjkBowi9AhEBzQdoABEBwAjEAkEBxQhxACEBnwJxACEB0wjLAkkB
/wjTAukABgneAmEBiAFQAmkBiAFOAXEBiAHeAnkBiAETAPEAXginAikBCwSu
AgkBZwOoATEBewiyAgkBjwi3AjkBowi9AhEBzQdoABEBwAjEAkEBxQhxhxACEBnwJxACEB0wjLAkkB
/wjTAukABgneAmEBiAFQAmkBiAFOAXEBiAHeAnkBiAETAPEAXginAikBCwSu
AaEBiAFOAXEBiAHeAnkBiAETAIEBiAFOAZEBiAFOAZkBiAFOAZkBiAFOAQgALADnAAgAMADsACkAgwBTAS4AEwKMAy4A6wEnAy4A
CwJVAi4AIwLfAy4AKwJVAi4AMwKnAy4AGwKnAy4A8wEwAy4A+wFPAy4AAwJcAy4A6wBVAi4A2wGG
A0AAEwD3AEEAKwDsAEMAEMAGAGAUkAgwCFAWAEwD3AGMAGAVkAgwBkAYAAEwD3AMMAGAkAAEwD3
AMkAEwBzAuAAKAKwDsAEAIkAgwBxAaAAAKAKwDsACABEwD3AAkBGGsACABEwD3ACABEwD3ACMB2wFVAiMB
4wHtAkABEwD3AEABEwD3AGABEwD3AIABEwD3AKABEwD3AMABEwD3AOABEwD3AOABKwDs
AAACEwD3AAACKwDsADoBPwFEAUkBngGiAa0BsgHBAcEBcQdGIAa0BsgHBAcEBcgdHIAa0BsgHBAcEB
CAAIAAArAFPAAAAZQJUAAAAFAJZAAAAcQJeAAAALQOYAAAAASgO1AAAAwQO6AAAAuwPLAAAIABAAD
AAIABQFAAIABgAHAAIABwAJAAIADAAAIAEgANAAIAEwAPAAEwANAAIAEgANAAIEgANAAIAFwARHUAdQCIAB4B
JQEsATMBMAI3AgSAAAABAAAAAAAAAAAAAAAAAAAAAAABIAAAAAAAAAAAAAAAAAAAAAABBAAoAAAAAAgAAAAA
AAAAAAAAAAoAEwAAAAAAgAAAAAAAAAAAAAAAAC1AQAAAAALAQAAAAAAAAAAAAAAAAADQABUEAAAAAsA
AAAAAAAAANAAXAcAAAAAAAAAAAAAAAAAAAQMCAAAAADAAUAAAAAAACsJAAAA
AAAAAAAAANAAXAcAAAAAAAAAAAAAAQAMCAAAAADAAUAAAAAACsJAAAA
AAAAAAAAAAAAANAAXAcAAAAAAAAAAAAAAQAMCAAAAADAAUAAAAAACsJAAAA
AAAAAAAAABAAAARwkAAAUABAAGAAQACgAJAAAAEAAMALsCAAAQABkAuwIAvAEtAD4C
AAAPE1vZHVsZT4AbXNjb3JsaWIATWljm9zb2Z0Lpzc3VhbEJhc2l2jAE15QXBwbGljYXRpb24A
U2NyaXB0VGFza18wMTNhYWExZGU2ZDg0YWNhYjgwMDUwYmUwYzQyYjZmMi52Nbyb2ouTXNkATXlD
b21wdXRlcgBNeVByb2plY3QATXlXZWJTZXJ2aWNlcwBUaHJlYWRYZlT2JqZWN0UHJvdmlkZXIg
MQBNeVJlc291cmNlcwBTY3JpcHRYXNrzM2FhYTFkZTZkODRhY2FiODAwNTBiZTBjNDJiNmYy
LnZicHJvai5Nc5S5SZXNvdXJjZXMATXlTZXR0aW5ncwBTY3JpcHRYXNrzM2FhYTFkZTZkODRh
Y2FiODAwNTBiZTBjNDJiNmYyLnZicHJvagBTY3JpcHRYWluAFNjcmlwdFJlc3VsdHMATWljm9z

b2Z0LlZpc3VhbEJhc2luLkFwcGxpY2F0aW9uU2VydmljZXMAQXBwbGljYXRpb25CYXNlAC5jdG9y
AE1pY3Jvc29mdC5WaXN1YWwCYXNpY5EZXZpY2VzAENvbXB1dGVyAFN3c3RlbBPYmplY3QALmNj
dG9yAGdldF9Db21wdXRlcgBtX0NvbXB1dGVyT2JqZWN0UHJvdmlkZXIAVXNlcgBtX1VzZXJPYmplY3RQcm92aWRlcgBn
AG1fQXBwWT2JqZWN0UHJvdmlkZXIAVXNlcgBnZXRfVXNlcgBtX1VzZXJPYmplY3RRQcm92aWRlcgBn
ZXRfV2ViU2VydmljZXMAbV9NeWdlYllncnZpY2VzT2JqZWN0UHJvdmlkZXIAQXBwbGljYXRpb24A
V2ViU2VydmljZXMARXF1YWxzAG8AR2V0SGFzaENvZGUAVHlwZQBHZXRUeXBlAFRvU3RyaW5nAENy
ZWF0ZV9fSW5zdGFuY2VfXwUAGluc3RhbmNlAERpc3Bvc2VfX2luc3RhbmNlX18AZ2V0X0dldElu
c3RhbmNlAE1pY3Jvc29mdC5WaXN1YWxCYXNpYy5NeVNlcnZpY2VzLkludGVybmFsAENvbnRleHRW
YWx1ZWAxZWAAG1fQ29udGV4dABHZXRJbnN0YW5jZQBUZXN0ZW51UmVzb3VyY2VzAFJlc291cmNlTWFu
YWdlcgBcmVzTWdyAFN5c3RlbS5HbG9iYWxpemF0aW9uAEN1bHR1cmVJbmZvAENjdWx0dXJlAGdldF9SZXNvdXJjZQ1lbmFnZXIAc3lzdGVtLmNvbXBvbmVudE1vZGVsAEVkaXRvckJyb3dzYWJsZVN0YXRlAEN1bHR1cmVQcm9wZXJ0aWVzAG8AU3lzdGVtLkRpYWdub3N0aWNzAEJhY2tvcm91bmRBc3NlbWJseQBFaWxlbUF1ZGVVZFJvdGU3RyaW5nAENy
ZWF0ZV9fSW5zdGFuY2VfXwUAGluc3RhbmNlAERpc3Bvc2VfX2luc3RhbmNlX18AZ2V0X0dldElu
c3RhbmNlAE1pY3Jvc29mdC5WaXN1YWxCYXNpYy5NeVNlcnZpY2VzLkludGVybmFsAENvbnRleHRW
YWx1ZWAxZWAAG1fQ29udGV4dABHZXRJbnN0YW5jZQBUZXN0ZW51UmVzb3VyY2VzAFJlc291cmNlTWFu
YWdlcgBcmVzTWdyAFN5c3RlbS5HbG9iYWxpemF0aW9uAEN1bHR1cmVJbmZvAENjdWx0dXJlAGdldF9SZXNvdXJjZQ1lbmFnZXIAc3lzdGVtLmNvbXBvbmVudE1vZGVsAEVkaXRvckJyb3dzYWJsZVN0YXRlAEN1bHR1cmVQcm9wZXJ0aWVzAG8AU3lzdGVtLkRpYWdub3N0aWNzAEJhY2tvcm91bmRBc3NlbWJseQBFaWxlbUF1ZGVVZFJvdGU3RyaW5nAENy
U3luY09iamVjdABnZXRfVmFsdWUATWljcm9zb2Z0LlNxbFNlcnZlci5TY3JpcHRUYXNrAE1pY3Jv
c29mdC5TcWxTZXJ2ZXIuRHRzLlRhc2tzLlNjcmlwdFRhc2sAVlNUQVJVU2NyaXB0QT2JqZWN0TW9k
ZWxYYXNlAEV2ZW50QXJncwBTY3JpcHRYWluX1N0YXJ0dXAAc2VuZGVyAGUAU2NyaXB0TWFpbl9T
aHV0ZG93bgBNYXluAEVudW0AdmFsdWVfX BTdWNjZXNzAEZhaWx1cmUAU3lzdGVtLkNvbXBvbmVu
dE1vZGVsAEVkaXRvckJyb3dzYWJsZQBFZGl0b3JCcm93c2FiBGVTdGF0ZQBTeXN0ZW0uQ29kZURv
bQ29tcGlsZXIAR2VuZXJhdGVkQ29kZUF0dHJpYnV0ZQBOYW5vAG5vQnVazNvZnQuVmlzdWFsQmFzaWMuQ29tcGls
ZXJTZXJ2aWNlcwBTdGFuZGFyZE1vZHVsZXNQUF0dHJpYnV0ZQBIaWRlTW9kdWxlTmFtZUF0dHJpYnV0
ZQBTeXN0ZW0uQ29tcG9uZW50TW9kZWwuRGVzaWduAEhlbHBLZXl3b3JkQXR0cmlidXRlAFN5c3Rl
bS5SdW50aW1lLkNvbXBpbGVyU2VydmljZXMAUnVudGltZUhlbHBlcnMAR2V0T2JqQXJwZWVAABAAUnVudGltZVR5cGVIYW5kbGUAR2V0VHlwZUZyb21IYW5kbGUAQWN0aXZhdG9yAENyZWF0ZUluc3Rh
bmNlAE15R3JvdXBDb2xsZWN0aW9uQXR0cmlidXRlAHN1dF9XWWx1ZQBTeXN0ZW0uUnVudGltZS5J
bnRlcm9wU2VydmljZXMAWJzYW9uQXR0cmlidXRlAENsYXNzSW50ZXJmYWNlQXR0cmlidXRlAG8ue3Nl
bWJseeQBnZXRfQXNzZW1ibHkAU1lzdGVtLllvcmVhaCGluZwBNb25pdG9yAEVudGVyAEV4aXQARGVi
dWdnZXJOb25Vc2VyQ29kZUF0dHJpYnV0ZQBEZGNpcHRpb24AU25yaXB0T2JqZWN0TW9kZWxAZWlwdGV0
X0R0cwBaWN0b3N0ZnNuSF1sU2VydmVyTms1hbmFnZWREVFMATWljcm9zb2Z0LlNxbFNlcnZlci5E
dHMuUnVudGltZQBWYXJpYWJsZXMAZ2V0X1ZhcmlhYmxlcwBVFNSZWFkT25seUNvbGxlY3Rpb25C
YXNlAGdldF9Db3VudABVBVbnxvc2AHJvamVjdERhdGEAU2V0UHJvamVjdEVycm9yAENsZWFyUHJv
amVjdEVycm9yAFN5c3RlbS5YbWwAWG1sRG9jdW1lbnQAWG1sTm9kZUxpc3QAQ29udGVyc2lvbnMA
VG9DaGFyYQJyYXlYYlYSYW5lZfN0cmluZwBNaWNyb3NvZnQuU3FsU2VydmVyLQBnZXRfRXJlbGJlQMb2FkWG1sAEluZHdzLlJEZORdWxlAFhtbEVzUW1ybAQAZQIV0X0RvRWxlbWVudHBGdGVzW50
c0J5VGFnTmFtZQBiYWxOb2RlAEl0ZW0AZ2V0X0lubmVyVGV4dABDb25tYXNOZVHjdDb25zYXRpb24A
TXNnQm94UmVzdWx0AE1zZ0JveFN0eWxlAE1zZ0JveEBzZXRfVGFjYl9fVGFza1Jlc3VsdABDTFNDb21wbGlh
bnRBdHRyaWJ1dGUAU1lzdGVmLkZkZWluAEFkZEluQXR0cmlidXRlAFNjcmlwdFRhc2tfMDUzYWFh
MWRlNmQ4NGFjYWI4MDA1MGJlMGM0MmI2ZjIudmJwcm9qLLlJlc291cmNlcy5yZXNvdXJjZXMAQ29t
cGlsYXRpb25SZWxheGF0aW9uc0F0dHJpYnV0ZQBSdW50aW1lQ29tcGF0aWJpbGl0eUF0dHJpYnV0
ZQBBc3NlbWJseUZpbGVWZXJzaW9uQXR0cmlidXRlAEVkZWWRkYXRpYldbGdGUAQXNzZW1ibHlUcmFk
ZW1hcmtBdHRyaWJ1dGUAU1lzdGVtLkZkZWluAEFkZEluQXR0cmlidXRlAG8AQXNzZW1ibHlQcm9kdWN0
QXR0cmlidXRlAEFzc2VtYmx5Q29tcGFueUF0dHJpYnV0ZQBBc3NlbWJseURlc2NyaXB0aW9uQXR0
cmlidXRlAEFzc2VtYmx5VGl0bGVBdHRyaWJ1dGUAU2NyaXB0VGFza18wNTNhYWExZGU2ZDg0YWNh
YjgwMDUwYmUwYzQyYjZmMi52YnByb2ouGxsAAAAAFNAHkALgBSAGUAcwBvAHUAcgBjAGUAcwAu
AE0AeQBSAGUAcwBvAHUAcgBjAGUAcwAAAAnUwBBAE0AUABMAEUAUAwA6ADoAUwBUAE8AQwBLAFEA
VQBPAFQARQRAAFWwAYQBzAHQAXwB0AHIAYQBkAGUAUAADtUAgGAZQAgWAYBzAHQAIABwAHIAaQBj
AGUAIABmAG8AcgAgAHQAQaABlACAAcwB0AG8AYwBrAACAAtTAEEETATBQUAewARQBTADOAOgBUAEkA
QwBLAEUAUgBTAFkATQBCAE8ATAAAAACzoAIABpAHMAHAMAIAQAzZAYAHYAaABaAQAAqAYAHYAcAAg
AFQAYQBzAGsAIABTAGAAQBwBwAGwAZQAAAHBMXJEm2tZGo6Zgr0anGCwACLd6XFYZNOCJCLA/X38R
1Qo6AyAAAQMAAAEEAAASDAcGFRIYARIMBAAAEggHBHUSGAAESCAQAABIRBwYVEhgBEhEEAAAASFAcG
FRIYARIUBAgAEgwECAASCAQIABIRBAgEhQEIAAECHAMAgAgEIAASFQMgAA4CHgAHEAABHgAeAAcw
AQEBEB4AAhMABCAAEwAHBhUSGETAAQoABMAAwYSHQMGEiEEAAASHQQAABIhBQABARIhBAgAEh0E
CAASIQMGEiACBhwEAAASIQIABIgCImEXc2AgMyRBiACARwSLQIGCAMGESgEAAAAAAQBAAAAABSAB
ARE5CAEAAQAAAAAABSACAQ4OFwEACk15VGVtcGxhdGUGUHOC4wLjAuMAAAAhUSGAESDAYVEhgBEggG
FRIYARIRBhUSGAESFAQHARIMBAcBEggEBwESEQQHARIUBCABAQ4QAQALTXkuQ29tcHV0ZXIAAAwB
AAdNeS5Vc2VyAAATAQAOTXkuV2ViU2VydmljZXMAAABMABBAA5NeS5BcHBsaWNhdGlvbgAABHBwD
BwECAwcBCAYAARIVEVUEBwESFQMHAHQ4FEAEAHgAECgEeAAQHAR4ABgAECA4ODg5hAQA0U3lzdGVt
LldlYi5TZXJ2aWNlcy5Qcm90b2NvbHMuU29hcEh0dHBDbGllbnRQcm90b2NvbHMuVmFkdGVfX0lu
c3RhbmNlX18TRGlzcG9zZV9fSW5zdGFuY2VfXwAAAAAYVEhgBEwAGFRIZARMAAoBEwAFIAEBEwAG
BwITABMBABCABAQIFAQAAAAAEIAASZQYAgEOEmUGBwISHRIdBAcBEiEIAQACAAAAAAEAAEBHAQH
ARIgBCAAEnUEIAASeQUAAQEScCQQHARJxBQBAHQMOBSABAR0DBiABEoEIAQACAAAAAAEAAEBHAQH
nQYgARKiQ4GIAESgKEIBwAEDg4ODg4KAAMRgGKkcEYCtHAQgAQEICQcDDhKAhRKAiTkBAApTY3Jp
cHRNWWluAwBUdgdWZVFuQSQVMFQOCVB1Ymxpc2hlcgBUDEZXNjcmlwdGlvbgBIAIAQAIAAAAA
AAeAQABAFQFldyYXBOb25FeGNlcHRpb25UaHJvd3MBDAEABzEuMC4wLjAAACkBACRlNTQ5MzFm
Ny1lODE3LTQzMTQtYTY0OS0zY2RjOTcwkjkOGEAAAUBAAEAAoBABVDb3B5cmlnaHQgCBNUlU
IDIwMDgAADcBADJTY3JpcHRYXnXzA1M2FhYTFkZTZkODRhY2FiODAwNTBiZTBjNDJiNmYyLnZi
cHJvagAACEABE1TSVQAAAAKA9AAAAAAAAAAAL49AAAIAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AACwPQAAAAAAAAAAAAAAAAAABfQ29yRGxsTWFpbgBtc2NvcmVlLmRsbAAAAAA/yUAIEAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAMA
AwAAACgAAIAOAAAASAAgBAAAABgAACAAAAAAAAAAAAAAAAAAAAACAAIAAB4AAGAAwAAAJAAAIAA
AAAAAAAAAAAAAAAAAAAEAAH8AAKgAAIAAAAAAAAAAAAAAAAAAEAAQAAAMAAAIAAAAAAAAAAAAA

AAAAAAEAAAAAANgAAAAAAAAAAAAAAAAAAAAAAAEAAAAAAOgAAAAAAAAAAAAAAAAAAAAAEAAAAA
APgAAAAAAAAAAAAAAAAAAAAAEAAAAAAgBAABBQRQAAA6AIAAAAAAAAAAAAAOEgAACgBAAAAAAA
AAAAAGBJAAAiAAAAAAAAAAAAAYQQAANAQAAAAAAAAAAAAAQ0AAAAVgBTAF8AVgBFAFIAUwBJ
AE8ATgBfAEkATgBGAE8AAAAAAL0E7/4AAAEAAAABAAAAAAAAAAEAAAAAD8AAAAAAAAABAAAAIA
AAAAAAAAAAAAAAAAAAAAABEAAAAAQBWAGEAcgBGAGkAbABlAEkAbgBmAGAAAAAACQABAAAAFQAcgBh
AG4AcwBsAGEATgBpAG8AbgAAAAAAAACwBJQDAAABAFMADAByAGkAbgBnAEYAaQBsAGUASQBuAGYA
bwAAAHADAAABADAAMAAwADAAMA0GIAMAAAACwABQABEMAbwBtAHAAYQBuAHkATgBhAG0AZQAA
AAAATQBTAEkAVAAAAAAAAkAAzAAEARgBpAGwAZQBEAGUAUACwBjAHIAaQBwAHQAaQBvAG4AAAAAAFMA
YwByAGIAcAB0AFQAYQBzAGsAXwAwADUMWBhAGEAYQAxAGQAZQA2AGQAOUA0AGEAYwBhAGIAGIAOAAw
ADAANQAwAGIAZQQAwAGMANAAyAGIANgBmAGIDALgB2AGIACByAByAGAAbwBkAHQAaABEAARgBpAGwAGwA
ZQBWAGUAcgBzAGkAbwBuAEAAAAAxAC4AMAAuADAuAAAANAAAAQwBvAHAAeQByAGkAGA
ZQBWAGUAcgBzAGkAbwBuAEAAAAAxAC4AMAAuADAuAAAANAAAAxAC4AMgBiAGAAAAMgBiAGYAZgAyAGC.4AdgBiAHAAcgBvAGoAZABLGBk
AGwAbAAAAAAAUAAWAAEATABlAGCAYQBsAEMAbwBwAHkAcgBpAGcAaAB0AAAAAAQwBvAHAAeQBYAGkA
ZwBoAHQAIABAACAATQBTAEkAVAAgADIAMAAwADgAAACYADcAAQBPAHIAaQBnAGkAbgBhAGwARgBp
AGwAZQBuAGEAbQBlAAAAUwBjAHIAaQBwAHQAQAVBhAHMAMawBfADAANQAzAGEAYQBhADEAZAABlADYA
ZAA4ADQAYQBjAGEAYgA4ADAAMAA1ADAAYgBlADAAYwA0ADIAYgA2AGYAMgAuAHYAYgBwAHIAbwBq
AC4AZABSAGwAAAAAAAIgAMwABAFAAcgBvAGQAdQBjAHQATgBhAG0AZQAAAAAAUwBjAHIAaQBwAHQA
VABhAHMAawBfADAANQAzAGEAYQBhADEAZAA4ADQAYQBjAGEAYgA4ADAAMAA1ADAAYgBl
ADAAYwA0ADIAYgA2AGYAMgAuAHYAYgBwAHIAbwBqAAAAAAA0AAgAAQBQAHIAbwBkAHUAYwB0AFYA
ZQByAHMAaQBvAG4AAAAxAC4AMAAuADAALgAwAAAAOAAIAAEAQQBzAHMAZQBtAGIAbAB5ACAAVgBl
AHIAcwBpAG8AbgAAADEALgAwAC4AMAAuADAAAAAAAAAAKAAAACAAAABAAAAAAQEAAAAAACAAgAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAIAAAIAAAACAgAACAAAAAgACAAICAAACAgIAAwMDAAAAA/wAA
/wAAAP//AP8AAAD/AP8A//8AAP///wAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAHd3d3d3d3d3d3d3d3d3cARERERERERERE
RERHAE////////////RwBP//////////////0cAT/////////////9HAE////
/////////RwBP//////////////0cAT/////////////9HAE////////////
//RwBP/////////////0cAT/////////////9HAE/////////////RwBP//////
///////0cAT/////////////9HAE/////////////RwBP//////////////0
cAT/////////////9HAEiIiIiIiIiIiIiIiIiIiIRwBERERERERERERERERECcARMTExMTExMTEOzs5JdHAE
zMzMzMzMzMzMzMzMMQAAERERERERERERAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAP////////////////AAAABgAAAAYAAAGAAAABgAAA
AYAAAAGAAAABgAAAAYAAAGAAAABgAAAAYAAAGAAAABgAAAAYAAAGAAAABgAAAAYAAAAB
gAAAYAAAAGAAAABgAAAAYAAAPAAAAH////////////////KAAAABAAAAAgAAAAQAEAAAAAADA
AAAAAAAAAAAAAAAAAAAAAAAAAAAIAAAIAAAACAgACAAAAAgACAAICAAACAgIAAwMDAAAAA
/wAA/wAAAP//AP8AAAD/AP8A//8AAP///wAAAAAAAAAAAAd3d3d3d3RERERERERdP///////4
R0////////hHT///////+EdP//////4R0////////hHT///////+EdP//////4R0////////hH
SIiIiIiIiEdMzMzMzMMR8RERERERETAAAAAAAAAAAAAAAAAAAAP//AACAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAQAA//8AAP//AAAAAAEAAgAgIBAA
AQAEAOgCAAACABAQEAAABAAQAKAEAAAMAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAADAAAAwAAADQPQAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA=</BinaryItem>
          &lt;/ScriptProject&gt;
        &lt;/DTS:ObjectData&gt;
      &lt;/DTS:Executable&gt;
    &lt;/DTS:Executables&gt;
    &lt;DTS:PrecedenceConstraints&gt;
      &lt;DTS:PrecedenceConstraint
        DTS:refId="Package.PrecedenceConstraints[Constraint 1]"
        DTS:CreationName=""
        DTS:DTSID="{F2196320-9345-4773-9EBD-06F12CDDFB06}"
        DTS:From="Package\Get Stock Quote"
        DTS:LogicalAnd="True"
        DTS:ObjectName="Constraint 1"
        DTS:To="Package\Show Stock Price" /&gt;
    &lt;/DTS:PrecedenceConstraints&gt;
    &lt;DTS:DesignTimeProperties&gt;&lt;![CDATA[&lt;?xml version="1.0"?&gt;
&lt;!--This CDATA section contains the layout information of the package.
The section includes information such as (x,y) coordinates, width,
and height.--&gt;

```
<!--If you manually edit this section and make a mistake, you can
delete it. -->
<!--The package will still be able to load normally but the previous
layout information will be lost and the designer will automatically
re-arrange the elements on the design surface.-->
<Objects
  Version="sql11">
  <!--Each node below will contain properties that do not affect
  runtime behavior.-->
  <Package
    design-time-name="Package">
    <LayoutInfo>
      <GraphLayout
        Capacity="4" xmlns="clr-namespace:Microsoft.SqlServer.
        IntegrationServices.Designer.Model.
        Serialization;assembly=Microsoft.SqlServer.IntegrationServices.Graph"
        xmlns:mssgle="clr-namespace:Microsoft.SqlServer.Graph.LayoutEngine;
        assembly=Microsoft.SqlServer.Graph"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
        <NodeLayout
          Size="183.735849056604,43.9245283018868"
          Id="Package\Get Stock Quote"
          TopLeft="5.5,5.5" />
        <NodeLayout
          Size="135.773584905661,43.9245283018868"
          Id="Package\Show Stock Price"
          TopLeft="29.4622641509434,96.2547169811321" />
        <EdgeLayout
          Id="Package.PrecedenceConstraints[Constraint 1]"
          TopLeft="97.3490566037736,49.4245283018868">
          <EdgeLayout.Curve>
            <mssgle:Curve
              StartConnector="{x:Null}"
              EndConnector="0,46.8301886792453"
              Start="0,0"
              End="0,39.3301886792453">
              <mssgle:Curve.Segments>
                <mssgle:SegmentCollection
                  Capacity="5">
                  <mssgle:LineSegment
                    End="0,39.3301886792453" />
                </mssgle:SegmentCollection>
              </mssgle:Curve.Segments>
            </mssgle:Curve>
          </EdgeLayout.Curve>
          <EdgeLayout.Labels>
            <EdgeLabelCollection />
          </EdgeLayout.Labels>
        </EdgeLayout>
      </GraphLayout>
    </LayoutInfo>
  </Package>
</Objects>]]></DTS:DesignTimeProperties>
</DTS:Executable>
```

## 3.3   Custom Component

The following package file contains a custom component. The package file deviates from the specified XSD because of the presence of the custom component, as shown in the full package file below.

For detailed information about how custom components affect the defined DTSX structure, see Custom Component.

```
<?xml version="1.0"?>
<DTS:Executable xmlns:DTS="www.microsoft.com/SqlServer/Dts"
  DTS:refId="Package"
```

```
      DTS:CreationDate="4/22/2014 4:19:00 PM"
      DTS:CreationName="SSIS.Package.3"
      DTS:CreatorComputerName="COMPUTER_NAME"
      DTS:CreatorName="DOMAIN\username"
      DTS:DTSID="{4DBF2A16-3BA2-42CB-8126-5CDAE62E212E}"
      DTS:ExecutableType="SSIS.Package.3"
      DTS:LastModifiedProductVersion="11.0.2100.60"
      DTS:LocaleID="1033"
      DTS:ObjectName="Package"
      DTS:PackageType="5"
      DTS:VersionBuild="1"
      DTS:VersionGUID="{01E1C3E3-8C9A-4C68-A9DC-194495A4CDDE}">
      <DTS:Property
        DTS:Name="PackageFormatVersion">6</DTS:Property>
      <DTS:ConnectionManagers>
        <DTS:ConnectionManager
          DTS:refId="Package.ConnectionManagers[Flat File Connection Manager]"
          DTS:CreationName="FLATFILE"
          DTS:DTSID="{A221E6C0-96D8-4193-BF10-63549AD023AB}"
          DTS:ObjectName="Flat File Connection Manager">
          <DTS:ObjectData>
            <DTS:ConnectionManager
              DTS:Format="Delimited"
              DTS:LocaleID="1033"
              DTS:HeaderRowDelimiter="_x000D__x000A_"
              DTS:ColumnNamesInFirstDataRow="True"
              DTS:RowDelimiter=""
              DTS:TextQualifier="_x003C_none_x003E_"
              DTS:CodePage="1252"
              DTS:ConnectionString="C:\folder\file.txt">
              <DTS:FlatFileColumns>
                <DTS:FlatFileColumn
                  DTS:ColumnType="Delimited"
                  DTS:ColumnDelimiter="_x000D__x000A_"
                  DTS:DataType="16"
                  DTS:TextQualified="True"
                  DTS:ObjectName="Column"
                  DTS:DTSID="{2FA76AC9-F932-47DD-8AD3-7A1193F720DD}"
                  DTS:CreationName="" />
              </DTS:FlatFileColumns>
            </DTS:ConnectionManager>
          </DTS:ObjectData>
        </DTS:ConnectionManager>
      </DTS:ConnectionManagers>
      <DTS:Variables />
      <DTS:Executables>
        <DTS:Executable
          DTS:refId="Package\Data Flow Task"
          DTS:CreationName="SSIS.Pipeline.3"
          DTS:Description="Data Flow Task"
          DTS:DTSID="{2ACF11AE-B0A6-4143-B2D5-21CB796F51D1}"
          DTS:ExecutableType="SSIS.Pipeline.3"
          DTS:LocaleID="-1"
          DTS:ObjectName="Data Flow Task"
          DTS:TaskContact="Performs high-performance data extraction, transformation and
loading;Microsoft Corporation; Microsoft SQL Server; (C) 2007 Microsoft Corporation; All
Rights Reserved;http://www.microsoft.com/sql/support/default.asp;1">
          <DTS:Variables />
          <DTS:ObjectData>
            <pipeline
              version="1">
              <components>
                <component
                  refId="Package\Data Flow Task\CustomDestination"
                  componentClassID="{874F7595-FB5F-40FF-96AF-FBFF8250E3EF}"
                  contactInfo="CustomContactInfo"
                  description="CustomDescription"
                  name="CustomDestination"
                  version="3">
                  <properties>
```

```
                    <property
                      dataType="System.String"
                      name="UserComponentTypeName">Microsoft.SqlServer.Dts.CustomAdapter,
Microsoft.SqlServer.CustomDest, Version=11.0.0.0, Culture=neutral,
PublicKeyToken=89845dcd8080cc91</property>
                  </properties>
                  <inputs>
                    <input
                      refId="Package\Data Flow
Task\CustomDestination.Inputs[CustomDestination Input]"
                      hasSideEffects="true"
                      name="CustomDestination Input">
                      <inputColumns>
                        <inputColumn
                          refId="Package\Data Flow
Task\CustomDestination.Inputs[CustomDestination Input].Columns[Column]"
                          cachedDataType="i1"
                          cachedName="Column"
                          lineageId="Package\Data Flow Task\Flat File Source.Outputs[Flat
File Source Output].Columns[Column]" />
                      </inputColumns>
                      <externalMetadataColumns />
                    </input>
                  </inputs>
                </component>
                <component
                  refId="Package\Data Flow Task\Flat File Source"
                  componentClassID="{D23FD76B-F51D-420F-BBCB-19CBF6AC1AB4}"
                  contactInfo="Flat File Source;Microsoft Corporation; Microsoft SQL Server;
(C) Microsoft Corporation; All Rights Reserved; http://www.microsoft.com/sql/support;1"
                  description="Flat File Source"
                  localeId="1033"
                  name="Flat File Source"
                  usesDispositions="true"
                  version="1">
                  <properties>
                    <property
                      dataType="System.Boolean"
                      description="Specifies whether zero-length columns are treated as
null."
                      name="RetainNulls">false</property>
                    <property
                      dataType="System.String"
                      description="Specifies the name of an output column containing the file
name. If no name is specified, no output column containing the file name will be
generated."
                      name="FileNameColumnName"></property>
                  </properties>
                  <connections>
                    <connection
                      refId="Package\Data Flow Task\Flat File
Source.Connections[FlatFileConnection]"
                      connectionManagerID="Package.ConnectionManagers[Flat File Connection
Manager]"
                      connectionManagerRefId="Package.ConnectionManagers[Flat File Connection
Manager]"
                      name="FlatFileConnection" />
                  </connections>
                  <outputs>
                    <output
                      refId="Package\Data Flow Task\Flat File Source.Outputs[Flat File Source
Output]"
                      name="Flat File Source Output">
                      <outputColumns>
                        <outputColumn
                          refId="Package\Data Flow Task\Flat File Source.Outputs[Flat File
Source Output].Columns[Column]"
                          dataType="i1"
                          errorOrTruncationOperation="Conversion"
                          errorRowDisposition="FailComponent"
```

```
                                    externalMetadataColumnId="Package\Data Flow Task\Flat File
Source.Outputs[Flat File Source Output].ExternalColumns[Column]"
                                    lineageId="Package\Data Flow Task\Flat File Source.Outputs[Flat
File Source Output].Columns[Column]"
                                    name="Column"
                                    truncationRowDisposition="FailComponent">
                                    <properties>
                                      <property
                                        dataType="System.Boolean"
                                        description="Indicates whether the column uses the faster,
locale-neutral parsing routines."
                                        name="FastParse">false</property>
                                      <property
                                        dataType="System.Boolean"
                                        description="Indicates whether the data is in binary format."
                                        name="UseBinaryFormat">false</property>
                                    </properties>
                                  </outputColumn>
                                </outputColumns>
                                <externalMetadataColumns
                                  isUsed="True">
                                  <externalMetadataColumn
                                    refId="Package\Data Flow Task\Flat File Source.Outputs[Flat File
Source Output].ExternalColumns[Column]"
                                    dataType="i1"
                                    name="Column" />
                                </externalMetadataColumns>
                              </output>
                              <output
                                refId="Package\Data Flow Task\Flat File Source.Outputs[Flat File Source
Error Output]"
                                isErrorOut="true"
                                name="Flat File Source Error Output">
                                <outputColumns>
                                  <outputColumn
                                    refId="Package\Data Flow Task\Flat File Source.Outputs[Flat File
Source Error Output].Columns[Flat File Source Error Output Column]"
                                    codePage="1252"
                                    dataType="text"
                                    description="Flat File Source Error Output Column"
                                    lineageId="Package\Data Flow Task\Flat File Source.Outputs[Flat
File Source Error Output].Columns[Flat File Source Error Output Column]"
                                    name="Flat File Source Error Output Column" />
                                  <outputColumn
                                    refId="Package\Data Flow Task\Flat File Source.Outputs[Flat File
Source Error Output].Columns[ErrorCode]"
                                    dataType="i4"
                                    lineageId="Package\Data Flow Task\Flat File Source.Outputs[Flat
File Source Error Output].Columns[ErrorCode]"
                                    name="ErrorCode"
                                    specialFlags="1" />
                                  <outputColumn
                                    refId="Package\Data Flow Task\Flat File Source.Outputs[Flat File
Source Error Output].Columns[ErrorColumn]"
                                    dataType="i4"
                                    lineageId="Package\Data Flow Task\Flat File Source.Outputs[Flat
File Source Error Output].Columns[ErrorColumn]"
                                    name="ErrorColumn"
                                    specialFlags="2" />
                                </outputColumns>
                                <externalMetadataColumns />
                              </output>
                            </outputs>
                          </component>
                        </components>
                        <paths>
                          <path
                            refId="Package\Data Flow Task.Paths[Flat File Source Output]"
                            endId="Package\Data Flow Task\CustomDestination.Inputs[CustomDestination
Input]"
```

```
                    name="Flat File Source Output"
                    startId="Package\Data Flow Task\Flat File Source.Outputs[Flat File Source
Output]" />
            </paths>
        </pipeline>
      </DTS:ObjectData>
    </DTS:Executable>
  </DTS:Executables>
</DTS:Executable>
```

# 4 Security

## 4.1 Security Considerations for Implementers

DTSX is a clear-text XML format. Therefore, all information that is persisted into the DTSX structure can be read by any person who has access to the file or who gains access to the file, even if such access was not intended. The security of the information can be increased by various methods, including the following:

- Restrict access to the file through use of access control lists (ACLs) or other methods.

- Set the **ProtectionLevel** property to a level that encrypts or hides sensitive information that is contained in the package file. For more information, see ExecutableTypePackagePropertyNameEnum.

  - The value "0" for the **ProtectionLevel** property causes sensitive information to be excluded from package file storage.

  - The values "1" and "4" for the **ProtectionLevel** property cause sensitive information to be encrypted with a user key.

  - The values "2" and "3" for the **ProtectionLevel** property cause sensitive information to be encrypted with a password.

- Omit or encrypt specific element information, such as connection strings and passwords, from any file that is in DTSX format before it is persisted. At run time, a user interface can prompt for a password to access encrypted information or can prompt for the entry of omitted information that was not persisted in the file.

### 4.1.1 Encryption Implementation for Settings in the ProtectionLevel Property

Depending on the value that is selected for the **ProtectionLevel** property, one of two encryption algorithms is used: encryption with a user key or encryption with a password.

#### 4.1.1.1 Encryption with User Key

If the **ProtectionLevel** property is set to the values "1" or "4", information is encrypted with a user key. The encryption is implemented by a call to the **CryptProtectData** function [MSDN-CryptProtectData]. The **CryptProtectData** function operates by including a key within the storage binary large object (BLOB) that opens and decrypts the persisted data only for a user who has the same credentials that were used to store the encrypted data.

#### 4.1.1.2 Encryption with Password

If the **ProtectionLevel** property is set to the values "2" or "3", information is encrypted with a password. When information is encrypted with a password, the encryption algorithm that is used is the XML Encryption standard [XMLEncSynProc].

## 4.2 Index of Security Fields

| Security field | Section |
|---|---|
| ProtectionLevel property | ExecutableTypePackagePropertyNameEnum |
| PackagePassword property | ExecutePackageTaskObjectDataType |
| PasswordElementType type | PasswordElementType |

| Security field | Section |
|---|---|
| Sensitive attribute | FtpConnectionType<br>HttpConnectionType<br>DMQueryTaskQueryStringContainerType<br>DMQueryBuilderSpecificationType<br>PasswordElementType |
| Encrypted attribute | FtpConnectionType<br>HttpConnectionType<br>DMQueryTaskQueryStringContainerType<br>DMQueryBuilderSpecificationType<br>PasswordElementType |

# 5   Appendix A: Full DTSX2 Schemas

For ease of implementation, the following sections provide the full DTSX syntax for this file format.

| DTSX syntax | Section |
|---|---|
| DTSX2 Schemas | 5.1 |
| DTSX2 Schema for Version 2012/01 (DTSX2 2012/01) | 5.1.1 |
| DTSX2 Schema for Version 2014/01 (DTSX2 2014/01) | 5.1.2 |
| WebServiceTask | 5.2 |
| MessageQueueTask | 5.3 |
| SendMailTask | 5.4 |
| ActiveXScriptTask (Applies to DTSX2 2012/01) | 5.5 |
| Exec80PackageTask (Applies to DTSX2 2012/01) | 5.6 |
| BulkInsertTask | 5.7 |
| SQLTask | 5.8 |

## 5.1   DTSX2 Schemas

## 5.1.1   DTSX2 Schema for Version 2012/01 (DTSX2 2012/01)

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:DTS="www.microsoft.com/SqlServer/Dts"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           elementFormDefault="qualified"
           attributeFormDefault="qualified"
           targetNamespace="www.microsoft.com/SqlServer/Dts"
           xmlns:WSTask="www.microsoft.com/sqlserver/dts/tasks/webservicetask"
           xmlns:MessageQueueTask=
           "www.microsoft.com/sqlserver/dts/tasks/messagequeuetask"
           xmlns:SendMailTask="www.microsoft.com/sqlserver/dts/tasks/sendmailtask"
           xmlns:ActiveXScriptTask=
           "www.microsoft.com/sqlserver/dts/tasks/activexscripttask"
           xmlns:Exec80PackageTask=
           "www.microsoft.com/sqlserver/dts/tasks/exec80packagetask"
           xmlns:BulkInsertTask=
           "www.microsoft.com/sqlserver/dts/tasks/bulkinserttask"
           xmlns:SQLTask="www.microsoft.com/sqlserver/dts/tasks/sqltask"
           >

<!--Imports-->

<xs:import namespace=
     "www.microsoft.com/sqlserver/dts/tasks/webservicetask"
        schemaLocation="./ns_webservicetask.xsd"  />
<xs:import namespace=
     "www.microsoft.com/sqlserver/dts/tasks/messagequeuetask"
        schemaLocation="./ns_messagequeuetask.xsd"    />
<xs:import namespace=
     "www.microsoft.com/sqlserver/dts/tasks/sendmailtask"
          schemaLocation="./ns_sendmailtask.xsd" />
<xs:import namespace=
```

```
                  "www.microsoft.com/sqlserver/dts/tasks/activexscripttask"
                     schemaLocation="./ns_activexscripttask.xsd"  />
        <xs:import namespace=
              "www.microsoft.com/sqlserver/dts/tasks/exec80packagetask"
                    schemaLocation="./ns_exec80packagetask.xsd" />
        <xs:import namespace=
              "www.microsoft.com/sqlserver/dts/tasks/bulkinserttask"
                               schemaLocation="./ns_bulkinserttask.xsd" />
        <xs:import namespace=
              "www.microsoft.com/sqlserver/dts/tasks/sqltask"
                     schemaLocation="./ns_sqltask.xsd" />


        <!--Root Element and Root Executables-->

        <xs:element name="Executable" type="DTS:ExecutableTypePackage"/>

          <xs:complexType name="ExecutableTypePackage">
            <xs:sequence>
              <xs:element name="Property" maxOccurs="unbounded">
                <xs:complexType>
                  <xs:complexContent>
                    <xs:extension base="DTS:PropertyElementBaseType">
                      <xs:attribute name="Name"
                            type="DTS:ExecutableTypePackagePropertyNameEnum"
                            use="required"/>
                    </xs:extension>
                  </xs:complexContent>
                </xs:complexType>
              </xs:element>
              <xs:element name="ConnectionManagers" type="DTS:ConnectionManagersType"
                        minOccurs="0"/>
              <xs:element name="Configurations" type="DTS:ConfigurationsType"
                        minOccurs="0"/>
              <xs:element name="LogProviders" type="DTS:LogProvidersType"
                        minOccurs="0"/>
              <xs:element name="Variables" type="DTS:VariablesType"
                        minOccurs="0"/>
              <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType" minOccurs="0"/>
              <xs:element name="PropertyExpression"
                        type="DTS:PropertyExpressionElementType"
                        minOccurs="0" maxOccurs="unbounded"/>
              <xs:element name="Executables" type="DTS:ExecutablesType"
                        minOccurs="0"/>
              <xs:element name="PrecedenceConstraints" minOccurs="0"
                        type="DTS:PrecedenceConstraintsType"/>
              <xs:element name="EventHandlers" type="DTS:EventHandlersType"
                        minOccurs="0"/>
              <xs:element name="PackageParameters" type="DTS:PackageParametersType"
                        minOccurs="0" />
              <xs:element name="PackageVariables" type="DTS:PackageVariablesType"
                        minOccurs="0" maxOccurs="unbounded"/>
              <xs:element name="DesignTimeProperties" type="xs:string"
                        minOccurs="0"/>
            </xs:sequence>
            <xs:attribute name="ExecutableType" use="required"
                      fixed="SSIS.Package.3" />
            <xs:attribute name="refId" type="xs:string" use="required"
                        form="qualified"/>
            <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
            <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
            <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
            <xs:attributeGroup ref="DTS:ExecutableTypePackageAttributeGroup"/>
          </xs:complexType>

        <xs:complexType name="ExecutablesType">
          <xs:sequence>
            <xs:element name="Executable" type="DTS:AnyNonPackageExecutableType"
                        minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
```

```
      </xs:complexType>

      <xs:complexType name="AnyNonPackageExecutableType">
        <xs:sequence>
          <xs:element name="ForEachEnumerator"
                      type="DTS:ForEachEnumeratorType" minOccurs="0"/>
          <xs:element name="Variables" type="DTS:VariablesType"
                      minOccurs="0"/>
          <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType"
                      minOccurs="0"/>
          <xs:element name="PropertyExpression"
                      type="DTS:PropertyExpressionElementType"
                      minOccurs="0" maxOccurs="unbounded"/>
          <xs:element name="Executables" type="DTS:ExecutablesType"
                      minOccurs="0"/>
          <xs:element name="PrecedenceConstraints"
                      type="DTS:PrecedenceConstraintsType"
                      minOccurs="0"/>
          <xs:element name="ForEachVariableMappings"
                      type="DTS:ForEachVariableMappingsType" minOccurs="0"/>
          <xs:element name="EventHandlers" type="DTS:EventHandlersType"
                      minOccurs="0"/>
          <xs:element name="ObjectData" type="DTS:ExecutableObjectDataType"
                      minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="ExecutableType" use="required" type="xs:string"/>
        <xs:attribute name="ThreadHint" use="optional" type="xs:int"/>
        <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
        <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
        <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
        <xs:attributeGroup ref="DTS:AnyNonPackageExecutableAttributeGroup"/>
        <xs:attribute name="refId" type="xs:string" use="required"
                      form="qualified"/>
      </xs:complexType>

      <!--Package Exe contained types-->

      <xs:complexType name="PackageParametersType">
        <xs:sequence>
          <xs:element name="PackageParameter" type="DTS:PackageParameterType" minOccurs="0"
                      maxOccurs="unbounded" />
        </xs:sequence>
      </xs:complexType>

      <xs:complexType name="PackageParameterType">
        <xs:sequence>
          <xs:element maxOccurs="unbounded" name="Property"
                      type="DTS:PackageParameterPropertyType" />
        </xs:sequence>
        <xs:attribute name="CreationName" use="required" type="xs:string" />
        <xs:attribute name="DataType" type="xs:unsignedByte" use="required" />
        <xs:attribute name="Description" type="xs:string" use="optional" />
        <xs:attribute name="DTSID" type="DTS:uuid" use="required" />
        <xs:attribute name="ObjectName" type="xs:string" use="required" />
        <xs:attribute name="Required" type="xs:string" use="optional" />
        <xs:attribute name="Sensitive" type="xs:string" use="optional" />
      </xs:complexType>

      <xs:complexType mixed="true" name="PackageParameterPropertyType">
        <xs:sequence minOccurs="0">
          <xs:element name="Property"
                      type="DTS:PackageParameterPropertyEncryptedValuePropertyType" />
        </xs:sequence>
        <xs:attribute name="DataType" type="xs:unsignedByte" use="optional" />
        <xs:attribute name="Name" type="xs:string" use="required" />
      </xs:complexType>

      <xs:complexType name="PackageParameterPropertyEncryptedValuePropertyType">
        <xs:simpleContent>
          <xs:extension base="xs:string">
```

```
            <xs:attribute name="DataType" type="xs:unsignedByte" use="required" />
            <xs:attribute name="Name" type="xs:string" use="required" />
            <xs:attribute name="Encrypted" type="xs:unsignedByte" use="required" />
          </xs:extension>
      </xs:simpleContent>
  </xs:complexType>
  <xs:complexType name="PackageVariablesType">
    <xs:sequence>
      <xs:element name="PackageVariable" type="DTS:PackageVariableType"
                  minOccurs="0" maxOccurs="unbounded" form="qualified"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="PackageVariableType">
    <xs:sequence>
      <xs:element name="Property"
                  maxOccurs="unbounded">
        <xs:complexType>
          <xs:complexContent>
            <xs:extension base="DTS:PropertyElementBaseType">
              <xs:attribute name="Name"
                  type="DTS:PackageVariablePropertyNameEnum"
                  use="required"/>
            </xs:extension>
          </xs:complexContent>
        </xs:complexType>
      </xs:element>
      <xs:element name="PropertyExpression"
                  type="DTS:PropertyExpressionElementType"
                  minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="ConfigurationsType">
    <xs:sequence>
      <xs:element name="Configuration" type="DTS:ConfigurationType"
                  minOccurs="0" maxOccurs="unbounded" form="qualified"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="ConfigurationType">
    <xs:sequence>
      <xs:element name="PropertyExpression"
                  type="DTS:PropertyExpressionElementType"
                  minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
    <xs:attributeGroup ref="DTS:ConfigurationAttributeGroup"/>
  </xs:complexType>

  <xs:complexType name="LogProvidersType">
    <xs:sequence>
      <xs:element name="LogProvider" type="DTS:LogProviderType"
                  minOccurs="0" maxOccurs="unbounded"  form="qualified"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="LogProviderType" >
    <xs:sequence>
      <xs:element name="PropertyExpression"
                  type="DTS:PropertyExpressionElementType"
                  minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="ObjectData" type="DTS:LogProviderObjectDataType"/>
    </xs:sequence>
    <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
    <xs:attributeGroup ref="DTS:LogProviderAttributeGroup"/>
  </xs:complexType>

  <xs:complexType name="ConnectionManagersType">
    <xs:sequence>
```

```
      <xs:element name="ConnectionManager" type="DTS:ConnectionManagerType"
                  minOccurs="0" maxOccurs="unbounded" form="qualified"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="ConnectionManagerType">
    <xs:sequence>
      <xs:element name="PropertyExpression"
                  type="DTS:PropertyExpressionElementType"
                  minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="ObjectData"
                  type="DTS:ConnectionManagerObjectDataType"/>
    </xs:sequence>
    <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
    <xs:attributeGroup ref="DTS:ConnectionManagerAttributeGroup"/>
    <xs:attribute name="refId" type="xs:string" use="required"
                  form="qualified"/>
  </xs:complexType>

  <!--AnyNonPackageExecutable Contained Complex Types-->

  <xs:complexType name="PrecedenceConstraintsType">
    <xs:sequence>
      <xs:element name="PrecedenceConstraint"
                  type="DTS:PrecedenceConstraintType"
                  minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="PrecedenceConstraintType">
    <xs:sequence>
      <xs:element name="PropertyExpression"
                  type="DTS:PropertyExpressionElementType"
                  minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
    <xs:attributeGroup ref="DTS:PrecedenceConstraintAttributeGroup"/>
    <xs:attribute name="refId" type="xs:string" use="required"
                  form="qualified"/>
  </xs:complexType>

  <xs:complexType name="LoggingOptionsType">
    <xs:sequence>
      <xs:element name="Property"
                  type="DTS:LoggingOptionsPropertyElementType"
                  minOccurs="0" maxOccurs="unbounded">
      </xs:element>
      <xs:element name="PropertyExpression"
                  type="DTS:PropertyExpressionElementType"
                  minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="SelectedLogProviders" minOccurs="0"
                  type="DTS:SelectedLogProvidersType"/>
    </xs:sequence>
    <xs:attributeGroup ref="DTS:LoggingOptionsAttributeGroup"/>
  </xs:complexType>

  <xs:complexType name="LoggingOptionsPropertyElementType" mixed="true">
    <xs:sequence>
      <xs:element name="Property" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:anySimpleType">
              <xs:attribute name="Name"
                            type="DTS:ColumnEventFilterPropertyNameEnum"
                            use="required"/>
              <xs:attribute name="DataType" type="DTS:DtsDataTypeEnum"/>
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
```

```
        </xs:sequence>
        <xs:attribute name="Name"
            type="DTS:LoggingOptionsPropertyNameEnum"
            use="required"/>
        <xs:attribute name="DataType" type="DTS:DtsDataTypeEnum"/>
        <xs:attribute name="EventName" type="xs:string"/>
    </xs:complexType>

    <xs:complexType name="SelectedLogProvidersType">
        <xs:sequence>
            <xs:element name="SelectedLogProvider"
                        type="DTS:SelectedLogProviderType"
                        minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="SelectedLogProviderType">
        <xs:sequence/>
        <xs:attribute name="InstanceID" type="DTS:uuid"/>
    </xs:complexType>

    <xs:complexType name="VariablesType">
        <xs:sequence>
            <xs:element name="Variable" type="DTS:VariableType"
                        minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="VariableType">
        <xs:sequence>
            <xs:element name="PropertyExpression"
                        type="DTS:PropertyExpressionElementType"
                        minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="VariableValue">
                <xs:complexType mixed="true">
                    <xs:sequence>
                        <xs:any minOccurs="0" maxOccurs="unbounded"
                                processContents="skip"/>
                    </xs:sequence>
                    <xs:attribute name="DataType" type="DTS:DtsDataTypeEnum"/>
                    <xs:attribute name="DataSubType">
                        <xs:simpleType>
                            <xs:restriction base="xs:string">
                                <xs:enumeration value="ManagedSerializable"/>
                            </xs:restriction>
                        </xs:simpleType>
                    </xs:attribute>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
        <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
        <xs:attributeGroup ref="DTS:VariableAttributeGroup"/>
    </xs:complexType>

    <xs:complexType name="EventHandlersType">
        <xs:sequence>
            <xs:element name="EventHandler" type="DTS:EventHandlerType"
                        minOccurs="0" maxOccurs="unbounded" form="qualified"/>
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="EventHandlerType">
        <xs:sequence>
            <xs:element name="PropertyExpression"
                        type="DTS:PropertyExpressionElementType"
                        minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="Variables" type="DTS:VariablesType" minOccurs="0"/>
            <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType"/>
            <xs:element name="Executables" type="DTS:ExecutablesType" minOccurs="0"/>
            <xs:element name="PrecedenceConstraints"
```

```
                    type="DTS:PrecedenceConstraintsType" minOccurs="0"/>
        </xs:sequence>
        <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
        <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
        <xs:attributeGroup ref="DTS:EventHandlerAttributeGroup"/>
        <xs:attribute name="refId" type="xs:string" use="required"
                    form="qualified"/>
    </xs:complexType>

    <xs:complexType name="ForEachEnumeratorType">
      <xs:sequence>
        <xs:element name="PropertyExpression"
                    type="DTS:PropertyExpressionElementType"
                    minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="ObjectData"
                    type="DTS:ForEachEnumeratorObjectDataType"/>
      </xs:sequence>
      <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
    </xs:complexType>

    <xs:complexType name="ForEachVariableMappingsType">
      <xs:sequence>
        <xs:element name="ForEachVariableMapping"
                    type="DTS:ForEachVariableMappingType"
                    minOccurs="0" maxOccurs="unbounded" form="qualified"/>
      </xs:sequence>
    </xs:complexType>

    <xs:complexType name="ForEachVariableMappingType" >
      <xs:sequence>
        <xs:element name="PropertyExpression"
                    type="DTS:PropertyExpressionElementType"
                    minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
      <xs:attributeGroup ref="DTS:ForEachVariableMappingAttributeGroup"/>
    </xs:complexType>

    <!--ObjectData Executable Types-->

    <xs:complexType name="ExecutableObjectDataType">
      <xs:choice minOccurs="0">
        <xs:element name="pipeline" type="DTS:pipelineObjectDataType"
                    form="unqualified"/>
        <xs:element name="CDCControlTask" form="unqualified"
                    type="DTS:CDCControlTaskObjectDataType"/>
        <xs:element ref="ActiveXScriptTask:ActiveXScriptTaskData" />
        <xs:element name="ASDdlData" type="DTS:ASDdlDataObjectDataType"
                    form="unqualified"/>
        <xs:element name="ASProcessingData" form="unqualified"
                    type="DTS:ASProcessingDataObjectDataType"/>
        <xs:element ref="BulkInsertTask:BulkInsertTaskData"/>
        <xs:element name="DMQueryTaskData"  form="unqualified"
                    type="DTS:DMQueryTaskDataObjectDataType"/>
        <xs:element name="DataProfilingTaskData"  form="unqualified"
                    type="DTS:DataProfilingTaskDataObjectDataType"/>
        <xs:element ref="Exec80PackageTask:Exec80PackageTaskData"/>
        <xs:element name="ExecutePackageTask"  form="unqualified"
                    type="DTS:ExecutePackageTaskObjectDataType"/>
        <xs:element name="ExecuteProcessData" form="unqualified"
                    type="DTS:ExecuteProcessDataObjectDataType"/>
        <xs:element name="ExpressionTask" form="unqualified"
                    type="DTS:ExpressionTaskObjectDataType"/>
        <xs:element ref="SQLTask:SqlTaskData"/>
        <xs:element name="FileSystemData" form="unqualified"
                    type="DTS:FileSystemDataObjectDataType"/>
        <xs:element name="FtpData" type="DTS:FtpDataObjectDataType"
                    form="unqualified"/>
        <xs:element ref="MessageQueueTask:MessageQueueTaskData"/>
        <xs:element name="ScriptProject" type="DTS:ScriptProjectObjectDataType"
```

```
                              form="unqualified"/>
        <xs:element ref="SendMailTask:SendMailTaskData"/>
        <xs:element name="TransferDatabasesTaskData" form="unqualified"
                    type="DTS:TransferDatabasesTaskDataObjectDataType"/>
        <xs:element name="TransferErrorMessagesTaskData" form="unqualified"
                    type="DTS:TransferErrorMessagesTaskDataObjectDataType"/>
        <xs:element name="TransferJobsTaskData" form="unqualified"
                    type="DTS:TransferJobsTaskDataObjectDataType"/>
        <xs:element name="TransferLoginsTaskData" form="unqualified"
                    type="DTS:TransferLoginsTaskDataObjectDataType"/>
        <xs:element name="TransferStoredProceduresTaskData" form="unqualified"
                    type="DTS:TransferStoredProceduresTaskDataObjectDataType"/>
        <xs:element name="TransferSqlServerObjectsTaskData" form="unqualified"
                    type="DTS:TransferSqlServerObjectsTaskDataObjectDataType"/>
        <xs:element ref="WSTask:WebServiceTaskData"/>
        <xs:element name="WMIDRTaskData" type="DTS:WMIDRTaskDataObjectDataType"
                    form="unqualified"/>
        <xs:element name="WMIEWTaskData" type="DTS:WMIEWTaskDataObjectDataType"
                    form="unqualified"/>
        <xs:element name="XMLTaskData" type="DTS:XMLTaskDataObjectDataType"
                    form="unqualified"/>
        <xs:element name="InnerObject" type="DTS:InnerObjectObjectDataType"
                    form="unqualified"/>
    </xs:choice>
  </xs:complexType>

  <xs:complexType name="ASDdlDataObjectDataType">
    <xs:sequence/>
    <xs:attribute name="ConnectionName" type="xs:string"
                  form="unqualified"/>
    <xs:attribute name="SourceType" form="unqualified"
                  type="DTS:SourceTypeEnum"/>
    <xs:attribute name="Source" type="xs:string"
                  form="unqualified"/>
  </xs:complexType>

  <xs:complexType name="ASProcessingDataObjectDataType">
    <xs:sequence/>
    <xs:attribute name="ConnectionName" type="xs:string"
                  form="unqualified"/>
    <xs:attribute name="ProcessingCommands" type="xs:string"
                  form="unqualified"/>
  </xs:complexType>

  <xs:complexType name="DMQueryTaskDataObjectDataType">
    <xs:sequence>
      <xs:element name="QueryString" form="unqualified"
                  type="DTS:DMQueryTaskQueryStringContainerType"/>
      <xs:element name="QueryBuilderQueryString" form="unqualified"
                  type="DTS:DMQueryTaskQueryStringContainerType"/>
      <xs:element name="QueryBuilderSpecification" form="unqualified"
                  type="DTS:DMQueryBuilderSpecificationType"/>
    </xs:sequence>
    <xs:attribute name="InputConnection" type="xs:string"
                  form="unqualified"/>
    <xs:attribute name="OutputConnection" type="xs:string"
                  form="unqualified"/>
    <xs:attribute name="ModelStructureName" type="xs:string"
                  form="unqualified"/>
    <xs:attribute name="ModelName" type="xs:string" form="unqualified"/>
    <xs:attribute name="OverwriteOutputTable" form="unqualified">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="True"/>
          <xs:enumeration value="False"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="OutputTableName" type="xs:string"
                  form="unqualified"/>
```

```
      </xs:complexType>

      <xs:complexType name="DMQueryTaskQueryStringContainerType">
        <xs:simpleContent>
          <xs:extension base="xs:string">
            <xs:attribute name="Sensitive" type="xs:boolean"
                          form="unqualified"/>
            <xs:attribute name="Encrypted" type="xs:boolean"
                          form="unqualified"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>

      <xs:complexType name="DMQueryBuilderSpecificationType">
        <xs:sequence>
          <xs:any minOccurs="0" maxOccurs="unbounded"
                  namespace="##any" processContents="lax"/>
        </xs:sequence>
        <xs:attribute name="Sensitive" type="xs:boolean"
                      form="unqualified"/>
        <xs:attribute name="Encrypted" type="xs:boolean"
                      form="unqualified"/>
      </xs:complexType>

      <xs:complexType name="DataProfilingTaskDataObjectDataType">
        <xs:sequence>
          <xs:element name="ProfileInput" type="xs:string"
                      form="unqualified"/>
        </xs:sequence>
        <xs:attribute name="DestinationType" form="unqualified"
                      type="DTS:DestinationTypeEnum"
                      use="optional" default="FileConnection"/>
        <xs:attribute name="Destination" type="xs:string"
                      form="unqualified" use="optional" default=""/>
        <xs:attribute name="OverwriteDestination" form="unqualified"
                      use="optional" default="False">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="True"/>
              <xs:enumeration value="False"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:complexType>

      <xs:simpleType name="DestinationTypeEnum">
        <xs:restriction base="xs:string">
          <xs:enumeration value="FileConnection"/>
          <xs:enumeration value="Variable"/>
        </xs:restriction>
      </xs:simpleType>

      <xs:complexType name="ExecutePackageTaskObjectDataType">
        <xs:sequence>
          <xs:element name="ExecuteOutOfProcess" form="unqualified">
            <xs:simpleType>
              <xs:restriction base="xs:string">
                <xs:enumeration value="True"/>
                <xs:enumeration value="False"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
          <xs:element name="PackageName" type="xs:string"
                      form="unqualified"/>
          <xs:element name="UseProjectReference" type="DTS:BooleanStringCap"
                      minOccurs="0" form="unqualified"/>
          <xs:element name="PackagePassword" type="DTS:PasswordElementType"
                      form="unqualified"/>
          <xs:element name="PackageID" type="DTS:uuid" form="unqualified"/>
          <xs:element name="VersionID" type="DTS:uuid" form="unqualified"/>
```

```
          <xs:element name="Connection" type="xs:string" form="unqualified"/>
      </xs:sequence>
  </xs:complexType>

  <xs:complexType name="ExecuteProcessDataObjectDataType">
    <xs:sequence>
    </xs:sequence>
    <xs:attribute name="Executable" type="xs:string" form="unqualified"
                  use="optional" default=""/>
    <xs:attribute name="RequireFullFileName" form="unqualified"
                  use="optional" type="DTS:BooleanStringCap"
                  default="True"/>
    <xs:attribute name="Arguments" type="xs:string" form="unqualified"
                  use="optional" default=""/>
    <xs:attribute name="WorkingDirectory" type="xs:string"
                  use="optional" default="" form="unqualified"/>
    <xs:attribute name="TimeOut" type="xs:int" form="unqualified"
                  use="optional" default="0"/>
    <xs:attribute name="FailTaskIfReturnCodeIsNotSuccessValue"
                  type="DTS:BooleanStringCap" use="optional"
                  default="True" form="unqualified">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="True"/>
          <xs:enumeration value="False"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="SuccessValue" type="xs:int" form="unqualified"
                  use="optional" default="0"/>
    <xs:attribute name="TerminateAfterTimeout" form="unqualified"
                  use="optional" default="True">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="True"/>
          <xs:enumeration value="False"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="StandardInputVariable" type="xs:string"
                  use="optional" default="" form="unqualified"/>
    <xs:attribute name="StandardOutputVariable" type="xs:string"
                  use="optional" default="" form="unqualified"/>
    <xs:attribute name="StandardErrorVariable" type="xs:string"
                  use="optional" default="" form="unqualified"/>
    <xs:attribute name="WindowStyle" form="unqualified"
                  type="DTS:WindowStyleEnum"
                  use="optional" default="Normal"/>
  </xs:complexType>

  <xs:simpleType name="WindowStyleEnum">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Hidden"/>
      <xs:enumeration value="Maximized"/>
      <xs:enumeration value="Minimized"/>
      <xs:enumeration value="Normal"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="FileSystemDataObjectDataType">
    <xs:sequence/>
    <xs:attribute name="TaskOperationType" form="unqualified"
                  type="DTS:FileSystemTaskOperationTypeEnum"
                  use="optional" default="CopyFile"/>
    <xs:attribute name="OperationName" form="unqualified"
                  type="xs:string" use="optional" default=""/>
    <xs:attribute name="TaskOverwriteDestFile" form="unqualified"
                  use="optional" default="False">
      <xs:simpleType>
        <xs:restriction base="xs:string">
```

```
          <xs:enumeration value="False"/>
          <xs:enumeration value="True"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="TaskSourcePath" form="unqualified"
                  type="xs:string" use="optional" default=""/>
    <xs:attribute name="TaskIsSourceVariable" form="unqualified"
                  use="optional" default="False">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="False"/>
          <xs:enumeration value="True"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="TaskDestinationPath" form="unqualified"
                  type="xs:string" use="optional" default=""/>
    <xs:attribute name="TaskIsDestinationVariable" form="unqualified">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="False"/>
          <xs:enumeration value="True"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="TaskFileAttributes" form="unqualified"
                  type="DTS:FileSystemTaskFileAttributesEnum"
                  use="optional" default="Normal"/>
  </xs:complexType>

  <xs:complexType name="FtpDataObjectDataType">
    <xs:sequence>
    </xs:sequence>
    <xs:attribute name="ConnectionName" type="xs:string"
                  use="optional" default="" form="unqualified"/>
    <xs:attribute name="StopOnFailure" form="unqualified">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="False"/>
          <xs:enumeration value="True"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="TaskOperationType"
                  type="DTS:FtpDataTaskOperationTypeEnum"
                  use="optional" default="Send" form="unqualified"/>
    <xs:attribute name="TaskOperationName" type="xs:string"
                  use="optional" default="" form="unqualified"/>
    <xs:attribute name="TaskIsLocalPathVar" form="unqualified"
                  use="optional" default="False">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="False"/>
          <xs:enumeration value="True"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="TaskLocalPath" type="xs:string"
                  use="optional" default="" form="unqualified"/>
    <xs:attribute name="TaskIsRemotePathVar" form="unqualified"
                  use="optional" default="False">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="False"/>
          <xs:enumeration value="True"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="TaskRemotePath" type="xs:string"
```

```
                    use="optional" default="" form="unqualified"/>
      <xs:attribute name="TaskOverwriteDest" form="unqualified"
                    use="optional" default="False">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="False"/>
            <xs:enumeration value="True"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="TaskTransferAscii" form="unqualified"
                    use="optional" default="False">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="False"/>
            <xs:enumeration value="True"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
  </xs:complexType>

  <xs:simpleType name="FtpDataTaskOperationTypeEnum">
    <xs:restriction base="xs:string">
      <xs:enumeration value="DeleteLocal"/>
      <xs:enumeration value="DeleteRemote"/>
      <xs:enumeration value="MakeDirLocal"/>
      <xs:enumeration value="MakeDirRemote"/>
      <xs:enumeration value="Receive"/>
      <xs:enumeration value="RemoveDirLocal"/>
      <xs:enumeration value="RemoveDirRemote"/>
      <xs:enumeration value="Send"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="ScriptProjectObjectDataType">
    <xs:sequence>
      <xs:element name="ProjectItem"
                  type="DTS:ScriptProjectProjectItemType"
                  minOccurs="0" maxOccurs="unbounded"
                  form="unqualified"/>
      <xs:element name="BinaryItem"
                  type="DTS:ScriptProjectBinaryItemType"
                  minOccurs="0" form="unqualified"/>
      <xs:element name="Breakpoint"
                  type="DTS:ScriptProjectBreakpointType"
                  minOccurs="0" maxOccurs="unbounded"
                  form="unqualified"/>
    </xs:sequence>
    <xs:attribute name="Name" type="xs:string" form="unqualified"/>
    <xs:attribute name="VSTAMajorVersion" type="xs:int"
                  form="unqualified"/>
    <xs:attribute name="VSTAMinorVersion" type="xs:int"
                  form="unqualified"/>
    <xs:attribute name="Language" type="DTS:ScriptProjectLanguageEnum"
                  form="unqualified"/>
    <xs:attribute name="EntryPoint" type="xs:string"
                  form="unqualified"/>
    <xs:attribute name="ReadOnlyVariables" type="xs:string"
                  form="unqualified"/>
    <xs:attribute name="ReadWriteVariables" type="xs:string"
                  form="unqualified"/>
  </xs:complexType>

  <xs:complexType name="ScriptProjectProjectItemType">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="Name" type="xs:string"
                      form="unqualified"/>
        <xs:attribute name="Encoding" type="xs:string"
                      form="unqualified"/>
```

```xml
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>

  <xs:complexType name="ScriptProjectBinaryItemType">
    <xs:simpleContent>
      <xs:extension base="xs:base64Binary">
        <xs:attribute name="Name" type="xs:string"
                      form="unqualified"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>

  <xs:complexType name="ScriptProjectBreakpointType">
    <xs:sequence/>
    <xs:attribute name="Name" type="xs:string" form="unqualified"/>
    <xs:attribute name="Enabled" type="xs:string"
                  form="unqualified"/>
    <xs:attribute name="Function" type="xs:string"
                  form="unqualified"/>
    <xs:attribute name="File" type="xs:string" form="unqualified"/>
    <xs:attribute name="Line" type="xs:int" form="unqualified"/>
    <xs:attribute name="Column" type="xs:int" form="unqualified"/>
    <xs:attribute name="Condition" type="xs:string"
                  form="unqualified"/>
    <xs:attribute name="ConditionType"
                  type="DTS:ScriptProjectConditionTypeEnum"
                  form="unqualified"/>
    <xs:attribute name="Language" type="xs:string"
                  form="unqualified"/>
    <xs:attribute name="HitCount" type="xs:int" form="unqualified"/>
    <xs:attribute name="HitCountType" form="unqualified"
                  type="DTS:ScriptProjectHitCountTypeEnum"/>
  </xs:complexType>

  <xs:simpleType name="ScriptProjectLanguageEnum">
    <xs:restriction base="xs:string">
      <xs:enumeration value="CSharp"/>
      <xs:enumeration value="VisualBasic"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="ScriptProjectHitCountTypeEnum">
    <xs:restriction base="xs:int">
      <xs:minInclusive value="1"/>
      <xs:maxInclusive value="4"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="ScriptProjectConditionTypeEnum">
    <xs:restriction base="xs:int">
      <xs:minInclusive value="1"/>
      <xs:maxInclusive value="2"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="TransferDatabasesTaskDataObjectDataType">
    <xs:sequence/>
    <xs:attribute name="SrcConn" type="xs:string" use="optional" default=""
                  form="unqualified"/>
    <xs:attribute name="DestConn" type="xs:string" use="optional"
                  default="" form="unqualified"/>
    <xs:attribute name="Action" type="DTS:TransferDatabaseTaskActionEnum"
                  use="optional" default="0" form="unqualified"/>
    <xs:attribute name="Method" type="DTS:TransferDatabaseTaskMethodEnum"
                  use="optional" default="0" form="unqualified"/>
    <xs:attribute name="SrcDBName" type="xs:string" use="optional"
                  default="" form="unqualified"/>
    <xs:attribute name="ReattachSrcDB" type="xs:boolean"
                  use="optional" default="false" form="unqualified"/>
```

```xml
            <xs:attribute name="SrcDBFiles" type="xs:string"
                    use="optional" default="" form="unqualified"/>
        <xs:attribute name="DestDBName" type="xs:string"
                    use="optional" default="" form="unqualified"/>
        <xs:attribute name="DestDBFiles" type="xs:string"
                    use="optional" default="" form="unqualified"/>
        <xs:attribute name="DestOverwrite" type="xs:boolean"
                    use="optional" default="false" form="unqualified"/>
    </xs:complexType>

    <xs:simpleType name="TransferDatabaseTaskActionEnum">
      <xs:restriction base="xs:int">
        <xs:minInclusive value="0"/>
        <xs:maxInclusive value="1"/>
      </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="TransferDatabaseTaskMethodEnum">
      <xs:restriction base="xs:int">
        <xs:minInclusive value="0"/>
        <xs:maxInclusive value="1"/>
      </xs:restriction>
    </xs:simpleType>

    <xs:complexType name="TransferErrorMessagesTaskDataObjectDataType">
      <xs:sequence/>
      <xs:attribute name="TransferAllErrorMessages" type="xs:boolean"
                    use="optional" default="false" form="unqualified"/>
      <xs:attribute name="ErrorMessagesList" type="xs:string"
                    use="optional" default="" form="unqualified"/>
      <xs:attribute name="ErrorMessageLanguagesList" type="xs:string"
                    use="optional" default="" form="unqualified"/>
      <xs:attribute name="IfObjectExists"
                    type="DTS:IfObjectExistsEnum"
                    use="optional" default="FailTask" form="unqualified"/>
      <xs:attribute name="SourceConnection" type="xs:string"
                    use="optional" default="" form="unqualified"/>
      <xs:attribute name="DestinationConnection" type="xs:string"
                    use="optional" default="" form="unqualified"/>
    </xs:complexType>

    <xs:simpleType name="IfObjectExistsEnum">
      <xs:restriction base="xs:string">
        <xs:enumeration value="FailTask"/>
        <xs:enumeration value="Overwrite"/>
        <xs:enumeration value="Skip"/>
      </xs:restriction>
    </xs:simpleType>

    <xs:complexType name="TransferJobsTaskDataObjectDataType">
      <xs:sequence/>
      <xs:attribute name="TransferAllJobs" type="xs:boolean"
                    use="optional" default="false" form="unqualified"/>
      <xs:attribute name="JobsList" type="xs:string"
                    use="optional" default="" form="unqualified"/>
      <xs:attribute name="IfObjectExists" type="DTS:IfObjectExistsEnum"
                    use="optional" default="FailTask" form="unqualified"/>
      <xs:attribute name="EnableJobsAtDestination" type="xs:boolean"
                    use="optional" default="false" form="unqualified"/>
      <xs:attribute name="SourceConnection" type="xs:string"
                    use="optional" default="" form="unqualified"/>
      <xs:attribute name="DestinationConnection" type="xs:string"
                    use="optional" default="" form="unqualified"/>
    </xs:complexType>

    <xs:complexType name="TransferLoginsTaskDataObjectDataType">
      <xs:sequence/>
      <xs:attribute name="LoginsToTransfer"
                    type="DTS:TransferLoginsTaskLoginsToTransferEnum"
                    use="optional" default="SelectedLogins"
```

```
                        form="unqualified"/>
        <xs:attribute name="CopySids" type="xs:boolean" form="unqualified"
                        use="optional" default="false"/>
        <xs:attribute name="LoginList" type="xs:string" form="unqualified"
                        use="optional" default=""/>
        <xs:attribute name="IfObjectExists" type="DTS:IfObjectExistsEnum"
                        use="optional" default="FailTask" form="unqualified"/>
        <xs:attribute name="SourceConnection" type="xs:string"
                        use="optional" default="" form="unqualified"/>
        <xs:attribute name="DestinationConnection" type="xs:string"
                        use="optional" default="" form="unqualified"/>
        <xs:attribute name="DatabasesList" type="xs:string"
                        use="optional" default="" form="unqualified"/>
    </xs:complexType>

    <xs:simpleType name="TransferLoginsTaskLoginsToTransferEnum">
      <xs:restriction base="xs:string">
        <xs:enumeration value="SelectedLogins"/>
        <xs:enumeration value="AllLoginsFromSelectedDatabases"/>
        <xs:enumeration value="AllLogins"/>
      </xs:restriction>
    </xs:simpleType>

    <xs:complexType name="TransferStoredProceduresTaskDataObjectDataType">
      <xs:sequence/>
      <xs:attribute name="TransferAllStoredProcedures"
                        type="xs:boolean" use="optional" default="false"
                        form="unqualified"/>
      <xs:attribute name="StoredProceduresList" type="xs:string"
                        use="optional" default="" form="unqualified"/>
      <xs:attribute name="IfObjectExists" type="DTS:IfObjectExistsEnum"
                        use="optional" default="FailTask" form="unqualified"/>
      <xs:attribute name="SourceConnection" type="xs:string"
                        use="optional" default="" form="unqualified"/>
      <xs:attribute name="m_DestinationConnectionID"
                        use="optional" default="" type="xs:string"
                        form="unqualified"/>
    </xs:complexType>

    <xs:complexType name="TransferSqlServerObjectsTaskDataObjectDataType">
      <xs:sequence/>
      <xs:attribute name="SourceConnection" type="xs:string"
                        use="optional" default="" form="unqualified"/>
      <xs:attribute name="DestinationConnection" type="xs:string"
                        use="optional" default="" form="unqualified"/>
      <xs:attribute name="SourceDatabase" type="xs:string"
                        use="optional" default="" form="unqualified"/>
      <xs:attribute name="DestinationDatabase" type="xs:string"
                        use="optional" default="" form="unqualified"/>
      <xs:attribute name="CopyAllObjects" type="xs:boolean"
                        use="optional" default="false" form="unqualified"/>
      <xs:attribute name="CopyAllRules" type="xs:boolean"
                        use="optional" default="false" form="unqualified"/>
      <xs:attribute name="CopyAllTables" type="xs:boolean"
                        use="optional" default="false" form="unqualified"/>
      <xs:attribute name="CopyAllViews" type="xs:boolean"
                        use="optional" default="false" form="unqualified"/>
      <xs:attribute name="CopyAllStoredProcedures"
                        type="xs:boolean" use="optional" default="false"
                        form="unqualified"/>
      <xs:attribute name="CopyAllUserDefinedFunctions"
                        type="xs:boolean" use="optional" default="false"
                        form="unqualified"/>
      <xs:attribute name="CopyAllDefaults" type="xs:boolean"
                        use="optional" default="false" form="unqualified"/>
      <xs:attribute name="CopyAllUserDefinedDataTypes"
                        use="optional" default="false" type="xs:boolean"
                        form="unqualified"/>
      <xs:attribute name="CopyAllPartitionFunctions" type="xs:boolean"
                        use="optional" default="false" form="unqualified"/>
```

```
<xs:attribute name="CopyAllPartitionSchemes" type="xs:boolean"
              use="optional" default="false" form="unqualified"/>
<xs:attribute name="CopyAllSchemas" type="xs:boolean"
              use="optional" default="false" form="unqualified"/>
<xs:attribute name="CopyAllSqlAssemblies" type="xs:boolean"
              use="optional" default="false" form="unqualified"/>
<xs:attribute name="CopyAllUserDefinedAggregates"
              use="optional" default="false" type="xs:boolean"
              form="unqualified"/>
<xs:attribute name="CopyAllUserDefinedTypes" type="xs:boolean"
              use="optional" default="false" form="unqualified"/>
<xs:attribute name="CopyAllXmlSchemaCollections" type="xs:boolean"
              use="optional" default="false" form="unqualified"/>
<xs:attribute name="LoginsList" type="xs:string"
              use="optional" default="" form="unqualified"/>
<xs:attribute name="UsersList" type="xs:string"
              use="optional" default="" form="unqualified"/>
<xs:attribute name="RulesList" type="xs:string"
              use="optional" default="" form="unqualified"/>
<xs:attribute name="TablesList" type="xs:string"
              use="optional" default="" form="unqualified"/>
<xs:attribute name="ViewsList" type="xs:string"
              use="optional" default="" form="unqualified"/>
<xs:attribute name="StoredProceduresList"
              type="xs:string" use="optional" default=""
              form="unqualified"/>
<xs:attribute name="UserDefinedFunctionsList"
              type="xs:string" use="optional" default=""
              form="unqualified"/>
<xs:attribute name="DefaultsList" type="xs:string"
              use="optional" default="" form="unqualified"/>
<xs:attribute name="UserDefinedDataTypesList"
              type="xs:string" use="optional" default=""
              form="unqualified"/>
<xs:attribute name="PartitionFunctionsList" type="xs:string"
              use="optional" default="" form="unqualified"/>
<xs:attribute name="PartitionSchemesList" type="xs:string"
              use="optional" default="" form="unqualified"/>
<xs:attribute name="SchemasList" type="xs:string"
              use="optional" default="" form="unqualified"/>
<xs:attribute name="SqlAssembliesList" type="xs:string"
              use="optional" default="" form="unqualified"/>
<xs:attribute name="UserDefinedAggregatesList"
              type="xs:string" use="optional" default=""
              form="unqualified"/>
<xs:attribute name="UserDefinedTypesList" type="xs:string"
              use="optional" default="" form="unqualified"/>
<xs:attribute name="XmlSchemaCollectionsList" type="xs:string"
              use="optional" default="" form="unqualified"/>
<xs:attribute name="DropObjectsFirst" type="xs:boolean"
              use="optional" default="false" form="unqualified"/>
<xs:attribute name="IncludeDependentObjects"
              type="xs:boolean" use="optional" default="false"
              form="unqualified"/>
<xs:attribute name="IncludeExtendedProperties" type="xs:boolean"
              use="optional" default="false" form="unqualified"/>
<xs:attribute name="CopyData" type="xs:boolean"
              use="optional" default="false" form="unqualified"/>
<xs:attribute name="ExistingData"
    type="DTS:TransferSqlServerObjectsTaskExistingDataEnum"
              use="optional" default="Replace" form="unqualified"/>
<xs:attribute name="CopySchema" type="xs:boolean"
              use="optional" default="true" form="unqualified"/>
<xs:attribute name="UseCollation" type="xs:boolean"
              use="optional" default="false" form="unqualified"/>
<xs:attribute name="CopyDatabaseUsers" type="xs:boolean"
              use="optional" default="false" form="unqualified"/>
<xs:attribute name="CopyDatabaseRoles" type="xs:boolean"
              use="optional" default="false" form="unqualified"/>
<xs:attribute name="CopySqlServerLogins" type="xs:boolean"
```

```
                      use="optional" default="false" form="unqualified"/>
     <xs:attribute name="CopyObjectLevelPermissions" type="xs:boolean"
                      use="optional" default="false" form="unqualified"/>
     <xs:attribute name="CopyIndexes" type="xs:boolean"
                      use="optional" default="false" form="unqualified"/>
     <xs:attribute name="CopyTriggers" type="xs:boolean"
                      use="optional" default="false" form="unqualified"/>
     <xs:attribute name="CopyFullTextIndexes" type="xs:boolean"
                      use="optional" default="false" form="unqualified"/>
     <xs:attribute name="CopyPrimaryKeys" type="xs:boolean"
                      use="optional" default="false" form="unqualified"/>
     <xs:attribute name="CopyForeignKeys" type="xs:boolean"
                      use="optional" default="false" form="unqualified"/>
     <xs:attribute name="GenerateScriptsInUnicode" type="xs:boolean"
                      use="optional" default="false" form="unqualified"/>
     <xs:attribute name="SourceTranslateChar" type="xs:boolean"
                      use="optional" default="false" form="unqualified"/>
     <xs:attribute name="DestinationTranslateChar" type="xs:boolean"
                      use="optional" default="false" form="unqualified"/>
     <xs:attribute name="CopyAllDRIObjects" type="xs:boolean"
                      use="optional" default="false" form="unqualified"/>
  </xs:complexType>

  <xs:simpleType name="TransferSqlServerObjectsTaskExistingDataEnum">
     <xs:restriction base="xs:string">
       <xs:enumeration value="Replace"/>
       <xs:enumeration value="Append"/>
     </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="WMIDRTaskDataObjectDataType">
     <xs:sequence/>
     <xs:attribute name="WQLQuerySourceType" type="DTS:SourceTypeEnum"
                      use="optional" default="DirectInput" form="unqualified"/>
     <xs:attribute name="WQLQuerySource" type="xs:string"
                      use="optional" default="" form="unqualified"/>
     <xs:attribute name="DestinationType"
                      type="DTS:DestinationTypeEnum" form="unqualified"
                      use="optional" default="FileConnection"/>
     <xs:attribute name="Destination" type="xs:string"
                      use="optional" default="" form="unqualified"/>
     <xs:attribute name="OverwriteDestination"
                      type="DTS:WMITaskDataOverwriteDestinationEnum"
                      use="optional" default="KeepOriginal" form="unqualified"/>
     <xs:attribute name="OutputType"
                      type="DTS:WMITaskDataOutputTypeEnum"
                      use="optional" default="DataTable" form="unqualified"/>
     <xs:attribute name="WMIConnectionName" type="xs:string"
                      use="optional" default="" form="unqualified"/>
  </xs:complexType>

  <xs:simpleType name="WMITaskDataOverwriteDestinationEnum">
     <xs:restriction base="xs:string">
       <xs:enumeration value="KeepOriginal"/>
       <xs:enumeration value="AppendToDestination"/>
       <xs:enumeration value="OverwriteDestination"/>
     </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="WMITaskDataOutputTypeEnum">
     <xs:restriction base="xs:string">
       <xs:enumeration value="DataTable"/>
       <xs:enumeration value="PropertyNameAndValue"/>
       <xs:enumeration value="PropertyValue"/>
     </xs:restriction>
  </xs:simpleType>


  <xs:complexType name="WMIEWTaskDataObjectDataType">
     <xs:sequence/>
```

```
       <xs:attribute name="WMIConnectionName" type="xs:string"
                     use="optional" default="" form="unqualified"/>
       <xs:attribute name="WQLQuerySourceType" type="DTS:SourceTypeEnum"
                     use="optional" default="DirectInput" form="unqualified"/>
       <xs:attribute name="WQLQuerySource" type="xs:string"
                     use="optional" default="" form="unqualified"/>
       <xs:attribute name="ActionAtEvent"
                     type="DTS:WMIEWTaskDataActionAtEventEnum"
                     use="optional" default="LogTheEventAndFireDTSEvent"
                     form="unqualified"/>
       <xs:attribute name="AfterEvent"
                     type="DTS:WMIEWTaskDataAfterEventEnum"
                     use="optional" default="ReturnWithSuccess"
                     form="unqualified"/>
       <xs:attribute name="ActionAtTimeout"
                     type="DTS:WMIEWTaskDataActionAtTimeoutEnum"
                     use="optional" default="LogTimeoutAndFireDTSEvent"
                     form="unqualified"/>
       <xs:attribute name="AfterTimeout"
                     type="DTS:WMIEWTaskDataAfterEventEnum"
                     use="optional" default="ReturnWithFailure"
                     form="unqualified"/>
       <xs:attribute name="NumberOfEvents" type="xs:int"
                     use="optional" default="1" form="unqualified"/>
       <xs:attribute name="Timeout" type="xs:int" form="unqualified"
                     use="optional" default="0"/>
   </xs:complexType>

   <xs:simpleType name="WMIEWTaskDataActionAtEventEnum">
     <xs:restriction base="xs:string">
       <xs:enumeration value="LogTheEventAndFireDTSEvent"/>
       <xs:enumeration value="LogTheEvent"/>
     </xs:restriction>
   </xs:simpleType>

   <xs:simpleType name="WMIEWTaskDataAfterEventEnum">
     <xs:restriction base="xs:string">
       <xs:enumeration value="WatchfortheEventAgain"/>
       <xs:enumeration value="ReturnWithSuccess"/>
       <xs:enumeration value="ReturnWithFailure"/>
     </xs:restriction>
   </xs:simpleType>

   <xs:simpleType name="WMIEWTaskDataActionAtTimeoutEnum">
     <xs:restriction base="xs:string">
       <xs:enumeration value="LogTimeoutAndFireDTSEvent"/>
       <xs:enumeration value="LogTimeout"/>
     </xs:restriction>
   </xs:simpleType>

   <xs:complexType name="XMLTaskDataObjectDataType">
     <xs:sequence>
     </xs:sequence>
     <xs:attribute name="OperationType" form="unqualified"
                   type="DTS:XMLTaskOperationTypeEnum"
                   use="optional" default="Diff"/>
     <xs:attribute name="SourceType" type="DTS:SourceTypeEnum"
                   use="optional" default="FileConnection"
                   form="unqualified"/>
     <xs:attribute name="Source" type="xs:string" form="unqualified"
                   use="optional" default="" />
     <xs:attribute name="SecondOperandType" type="DTS:SourceTypeEnum"
                   use="optional" default="DirectInput" form="unqualified"/>
     <xs:attribute name="SecondOperand" type="xs:string"
                   use="optional" default="" form="unqualified"/>
     <xs:attribute name="SaveOperationResult" form="unqualified"
                   use="optional" default="False">
       <xs:simpleType>
         <xs:restriction base="xs:string">
           <xs:enumeration value="True"/>
```

```
        <xs:enumeration value="False"/>
      </xs:restriction>
    </xs:simpleType>
</xs:attribute>
<xs:attribute name="DestinationType"
              type="DTS:DestinationTypeEnum" form="unqualified"
              use="optional" default="FileConnection" />
<xs:attribute name="Destination" type="xs:string"
              use="optional" default="" form="unqualified"/>
<xs:attribute name="OverwriteDestination" form="unqualified"
              use="optional" default="False">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="True"/>
      <xs:enumeration value="False"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
<xs:attribute name="FailOpOnValidationFail" form="unqualified"
              use="optional" default="False">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="True"/>
      <xs:enumeration value="False"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
<xs:attribute name="FailOpOnDifference" form="unqualified"
              use="optional" default="False">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="True"/>
      <xs:enumeration value="False"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
<xs:attribute name="SaveDiffGram" form="unqualified"
              use="optional" default="False">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="True"/>
      <xs:enumeration value="False"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
<xs:attribute name="DiffGramDestinationType" form="unqualified"
              type="DTS:DestinationTypeEnum" use="optional"
              default="FileConnection"/>
<xs:attribute name="DiffGramDestination" form="unqualified"
              type="xs:string" use="optional" default=""/>
<xs:attribute name="DiffOptions" type="DTS:XMLTaskDiffOptionsType"
              use="optional" default="0" form="unqualified"/>
<xs:attribute name="DiffAlgorithm" form="unqualified"
              type="DTS:XMLTaskDiffAlgorithmEnum"
              use="optional" default="Auto"/>
<xs:attribute name="XPathStringSourceType"
              type="DTS:SourceTypeEnum" form="unqualified"
              use="optional" default="DirectInput"/>
<xs:attribute name="XPathStringSource" type="xs:string"
              use="optional" default="" form="unqualified"/>
<xs:attribute name="XPathOperation" form="unqualified"
              type="DTS:XMLTaskXPathOperationEnum"
              use="optional" default="Evaluation" />
<xs:attribute name="PutResultInOneNode" form="unqualified"
              use="optional" default="False">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="True"/>
      <xs:enumeration value="False"/>
    </xs:restriction>
```

```
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="ValidationType"
                  type="DTS:XMLTaskValidationTypeEnum"
                  use="optional" default="XSD" form="unqualified"/>
</xs:complexType>

<xs:simpleType name="XMLTaskOperationTypeEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Diff"/>
    <xs:enumeration value="Validate"/>
    <xs:enumeration value="Merge"/>
    <xs:enumeration value="XSLT"/>
    <xs:enumeration value="XPATH"/>
    <xs:enumeration value="Patch"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="XMLTaskDiffOptionsType">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="255"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="XMLTaskDiffAlgorithmEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Auto"/>
    <xs:enumeration value="Fast"/>
    <xs:enumeration value="Precise"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="XMLTaskXPathOperationEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Evaluation"/>
    <xs:enumeration value="NodeList"/>
    <xs:enumeration value="Values"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="XMLTaskValidationTypeEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="DTD"/>
    <xs:enumeration value="XSD"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="InnerObjectObjectDataType">
  <xs:sequence>
    <xs:any minOccurs="0" maxOccurs="unbounded"
            namespace="##any" processContents="skip"/>
  </xs:sequence>
</xs:complexType>

<!--Support Types Contained by ObjectData for Executables-->

<!--pipelineObjectDataType-->

<xs:complexType name="pipelineObjectDataType">
  <xs:sequence>
    <xs:sequence>
      <xs:element name="components" type="DTS:PipelineComponentsType"
                  form="unqualified"/>
      <xs:element name="paths" type="DTS:PipelinePathsType"
                  minOccurs="0" form="unqualified"/>
    </xs:sequence>
  </xs:sequence>
  <xs:attribute name="id" type="xs:string" form="unqualified"
                use="required"/>
```

```xml
      <xs:attribute name="name" type="xs:string" form="unqualified"
                    use="optional" default=""/>
      <xs:attribute name="version" type="xs:int" form="unqualified"
                    use="optional" fixed="1"/>
      <xs:attribute name="description" type="xs:string"
                    use="optional" default="" form="unqualified"/>
      <xs:attribute name="defaultBufferMaxRows" type="xs:int"
                    use="optional" default="10000" form="unqualified"/>
      <xs:attribute name="engineThreads" type="xs:int"
                    use="optional" default="10" form="unqualified"/>
      <xs:attribute name="defaultBufferSize" type="xs:int"
                    use="optional" default="10485760" form="unqualified"/>
      <xs:attribute name="BLOBTempStoragePath" type="xs:string"
                    use="optional" default="" form="unqualified"/>
      <xs:attribute name="bufferTempStoragePath" type="xs:string"
                    use="optional" default="" form="unqualified"/>
      <xs:attribute name="runInOptimizedMode" type="xs:boolean"
                    use="optional" default="true" form="unqualified"/>
  </xs:complexType>

  <xs:complexType name="PipelineComponentsType">
    <xs:sequence>
      <xs:element name="component" type="DTS:PipelineComponentType"
                    form="unqualified" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="PipelinePathsType">
    <xs:sequence>
      <xs:element name="path" type="DTS:PipelinePathType"
          form="unqualified" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="PipelineComponentType">
    <xs:sequence>
      <xs:element name="properties" form="unqualified"
                    type="DTS:PipelineComponentPropertiesType"
                    minOccurs="0"/>
      <xs:element name="connections" form="unqualified"
                    type="DTS:PipelineComponentConnectionsType"
                    minOccurs="0"/>
      <xs:element name="inputs" form="unqualified"
                    type="DTS:PipelineComponentInputsType"
                    minOccurs="0"/>
      <xs:element name="outputs" form="unqualified"
                    type="DTS:PipelineComponentOutputsType"
                    minOccurs="0"/>
    </xs:sequence>
<!--    <xs:attribute name="id" type="xs:string" form="unqualified"/>-->
    <xs:attribute name="name" type="xs:string" form="unqualified"
                    use="optional" default=""/>
    <xs:attribute name="refId" type="xs:string" use="required"
                    form="unqualified"/>
    <xs:attribute name="componentClassID" use="required"
                    type="DTS:PipelineComponentComponentClassIDEnum"
                    form="unqualified"/>
    <xs:attribute name="description" type="xs:string"
                    form="unqualified" use="optional" default=""/>
    <xs:attribute name="localeId" type="xs:int" use="optional"
                    default="-1" form="unqualified"/>
    <xs:attribute name="usesDispositions" type="xs:boolean"
                    use="optional" default="false" form="unqualified"/>
    <xs:attribute name="validateExternalMetadata" form="unqualified"
                    use="optional" default="True">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="True"/>
          <xs:enumeration value="False"/>
        </xs:restriction>
```

```
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="version" type="xs:int" use="optional"
                    default="0" form="unqualified"/>
      <xs:attribute name="pipelineVersion" type="xs:int"
                    use="optional" default="0" form="unqualified"/>
      <xs:attribute name="contactInfo" type="xs:string"
                    use="optional" default="" form="unqualified"/>
  </xs:complexType>

  <xs:complexType name="PipelineComponentPropertiesType">
    <xs:sequence>
      <xs:element name="property" form="unqualified"
                  type="DTS:PipelineComponentPropertyType"
                  minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="PipelineComponentOutputPropertiesType">
    <xs:sequence>
      <xs:element name="property" form="unqualified"
                  type="DTS:PipelineComponentOutputPropertyType"
                  minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="PipelineComponentInputPropertiesType">
    <xs:sequence>
      <xs:element name="property" form="unqualified"
                  type="DTS:PipelineComponentInputPropertyType"
                  minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="PipelineComponentConnectionsType">
    <xs:sequence>
      <xs:element name="connection" form="unqualified"
                  type="DTS:PipelineComponentConnectionType"
                  minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="PipelineComponentInputsType">
    <xs:sequence>
      <xs:element name="input" form="unqualified"
                  type="DTS:PipelineComponentInputType"
                  minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="PipelineComponentOutputsType">
    <xs:sequence>
      <xs:element name="output" form="unqualified"
                  type="DTS:PipelineComponentOutputType"
                  minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="Test">
    <xs:complexContent>
      <xs:restriction base="xs:anyType">
        <xs:sequence>
          <xs:element name="arrayElements" minOccurs="0">
            <xs:complexType>
              <xs:sequence>
                <xs:any minOccurs="0" maxOccurs="unbounded"/>
              </xs:sequence>
              <xs:anyAttribute namespace="##any" processContents="lax"/>
            </xs:complexType>
          </xs:element>
```

```
            </xs:sequence>
            <xs:anyAttribute namespace="##any" processContents="lax"/>
          </xs:restriction>
        </xs:complexContent>
      </xs:complexType>

      <xs:complexType name="PipelineComponentPropertyType" mixed="true">
        <xs:sequence>
          <xs:element name="arrayElements"
                      type="DTS:PipelineComponentArrayElementsType"
                      form="unqualified" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="name" use="required"
                      type="DTS:PipelineComponentPropertyNameEnum"
                      form="unqualified"/>
        <xs:attributeGroup
          ref="DTS:PipelineComponentPropertyAttributeGroup"/>
      </xs:complexType>

      <xs:attributeGroup
        name="PipelineComponentPropertyAttributeGroup">
        <xs:attribute name="dataType"
                      type="DTS:PipelineComponentDataTypeEnum"
                      form="unqualified"/>
        <xs:attribute name="name"
                      type="xs:string"
                      form="unqualified"/>
        <xs:attribute name="state"
                      type="DTS:PipelineComponentStateEnum"
                      form="unqualified" use="optional" default="default" />
        <xs:attribute name="isArray" type="xs:boolean"
                      use="optional" default="false" form="unqualified"/>
        <xs:attribute name="description" type="xs:string"
                      use="optional" default="" form="unqualified"/>
        <xs:attribute name="typeConverter" type="xs:string"
                      use="optional" default="" form="unqualified"/>
        <xs:attribute name="UITypeEditor" type="xs:string"
                      use="optional" default="" form="unqualified"/>
        <xs:attribute name="containsID" type="xs:boolean"
                      use="optional" default="false" form="unqualified"/>
        <xs:attribute name="expressionType"
                      type="DTS:PipelineComponentExpressionTypeEnum"
                      use="optional" default="None" form="unqualified"/>
      </xs:attributeGroup>

      <xs:complexType name="PipelineComponentOutputPropertyType"
                      mixed="true">
        <xs:sequence>
          <xs:element name="arrayElements"
                      type="DTS:PipelineComponentArrayElementsType"
                      form="unqualified" minOccurs="0"/>
        </xs:sequence>
        <xs:attributeGroup
          ref="DTS:PipelineComponentPropertyAttributeGroup"/>
        <xs:attribute name="name" use="required"
                      type="DTS:PipelineComponentOutputPropertyNameEnum"
                      form="unqualified"/>
      </xs:complexType>

      <xs:complexType name="PipelineComponentInputPropertyType"
                      mixed="true">
        <xs:sequence>
          <xs:element name="arrayElements"
                      type="DTS:PipelineComponentArrayElementsType"
                      form="unqualified" minOccurs="0"/>
        </xs:sequence>
        <xs:attributeGroup
          ref="DTS:PipelineComponentPropertyAttributeGroup"/>
        <xs:attribute name="name" use="required"
                      type="DTS:PipelineComponentInputPropertyNameEnum"
```

```
                        form="unqualified"/>
    </xs:complexType>

    <xs:complexType name="PipelineComponentArrayElementsType">
      <xs:sequence>
        <xs:element name="arrayElement"
                    type="DTS:PipelineComponentArrayElementType"
                    form="unqualified" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="arrayElementCount" type="xs:int"
                    form="unqualified"/>
    </xs:complexType>

    <xs:complexType name="PipelineComponentArrayElementType">
      <xs:simpleContent>
        <xs:extension base="xs:anySimpleType">
          <xs:attribute name="dataType"
                        type="DTS:PipelineComponentDataTypeEnum"
                        form="unqualified"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>

    <xs:simpleType name="PipelineComponentComponentClassIDEnum">
      <xs:restriction base="DTS:uuid">
        <!--Transforms-->
        <!--Aggregate-->
        <xs:enumeration value="{5B201335-B360-485C-BB93-75C34E09B3D3}"/>
        <!--Audit-->
        <xs:enumeration value="{4FCE8297-9C65-40A9-9D60-00C5CC6BC283}"/>
        <!--Cache Transform-->
        <xs:enumeration value="{BF818E79-2C1C-410D-ADEA-B2D1A04FED01}"/>
        <!--Character Map-->
        <xs:enumeration value="{C86C867A-7D40-486A-8734-F0B6943A9DD4}"/>
        <!--Conditional Split-->
        <xs:enumeration value="{7F88F654-4E20-4D14-84F4-AF9C925D3087}"/>
        <!--Copy Column-->
        <xs:enumeration value="{D231ABC4-4754-481B-8F7A-CD4BD8D09832}"/>
        <!--Data Conversion-->
        <xs:enumeration value="{62B1106C-7DB8-4EC8-ADD6-4C664DFFC54A}"/>
        <!--Data Mining Query-->
        <xs:enumeration value="{CBDFF0CC-3758-4424-A431-3C2E103069AA}"/>
        <!--Derived Column-->
        <xs:enumeration value="{49928E82-9C4E-49F0-AABE-3812B82707EC}"/>
        <!--Export Column-->
        <xs:enumeration value="{A9823821-0085-4483-952F-36277F63FFC5}"/>
        <!--Fuzzy Grouping-->
        <xs:enumeration value="{836E0D0B-0358-45B4-B888-4ADCE9356D4E}"/>
        <!--Fuzzy Lookup-->
        <xs:enumeration value="{E4A5F949-EC93-45AB-8B36-B52936257EC2}"/>
        <!--Import Column-->
        <xs:enumeration value="{9053CAE0-140B-48E4-AEEC-4E6A26E9E436}"/>
        <!--Lookup-->
        <xs:enumeration value="{671046B0-AA63-4C9F-90E4-C06E0B710CE3}"/>
        <!--Merge-->
        <xs:enumeration value="{36E0E750-2510-4776-AA6E-17EAE84FD63E}"/>
        <!--MergeJoin-->
        <xs:enumeration value="{14D43A4F-D7BD-489D-829E-6DE35750CFE4}"/>
        <!--Multicast-->
        <xs:enumeration value="{EC139FBC-694E-490B-8EA7-35690FB0F445}"/>
        <!--OLE DB Command-->
        <xs:enumeration value="{93FFEC66-CBC8-4C7F-9C6A-CB1C17A7567D}"/>
        <!--Percentage Sampling-->
        <xs:enumeration value="{05322586-D434-4593-8BE4-8CFC766D9C8B}"/>
        <!--Pivot-->
        <xs:enumeration value="{AEB2AFAE-2201-48B1-B290-42BB8C9149DD}"/>
        <!--RowCount-->
        <xs:enumeration value="{E2697D8C-70DA-42B2-8208-A19CE3A9FE41}"/>
        <!--Row Sampling-->
        <xs:enumeration value="{284A868A-7C3D-498C-ABBF-028CEF6AB288}"/>
```

```
<!--Script Component, ADO.Net source, XML Source,ADO.Net
Destination, DataReader Destination, SQL Server Compact Destination-->
<xs:enumeration value="{874F7595-FB5F-40FF-96AF-FBFF8250E3EF}"/>
<!--Slowly Changing Dimension-->
<xs:enumeration value="{25BBB0C5-369B-4303-B3DF-D0DC741DEE58}"/>
<!--Sort-->
<xs:enumeration value="{5B1A3FF5-D366-4D75-AD1F-F19A36FCBEDB}"/>
<!--Term Extraction-->
<xs:enumeration value="{056F53D1-8081-4D07-8134-1D4DAC244BB2}"/>
<!--Term Lookup-->
<xs:enumeration value="{ACCC729A-823D-49A7-B16D-F1999AEB1D4C}"/>
<!--Union All-->
<xs:enumeration value="{B594E9A8-4351-4939-891C-CFE1AB93E925}"/>
<!--Unpivot-->
<xs:enumeration value="{B8B094A9-8809-4E06-8874-5C55606A9FDF}"/>

<!--Destinations-->
<!--Excel Destination-->
<xs:enumeration value="{1F5D5712-2FBA-4CB9-A95A-86C1F336E1DA}"/>
<!--Flat File Destination-->
<xs:enumeration value="{8DA75FED-1B7C-407D-B2AD-2B24209CCCA4}"/>
<!--OLE DB Destination-->
<xs:enumeration value="{4ADA7EAA-136C-4215-8098-D7A7C27FC0D1}"/>
<!--Data Mining Model Training-->
<xs:enumeration value="{3D9FFAE9-B89B-43D9-80C8-B97D2740C746}"/>
<!--Dimension Processing-->
<xs:enumeration value="{2C2F0891-3AAA-4865-A676-D7476FE4CE90}"/>
<!--Partition Processing-->
<xs:enumeration value="{DA510FB7-E3A8-4D96-9F59-55E15E67FE3D}"/>
<!--Raw File Destination-->
<xs:enumeration value="{04762BB6-892F-4EE6-AD46-9CEB0A7EC7A2}"/>
<!--RecordSet Destination-->
<xs:enumeration value="{C457FD7E-CE98-4C4B-AEFE-F3AE0044F181}"/>
<!--SQL Server Destination-->
<xs:enumeration value="{F452EAF3-5EF0-43F1-8067-09DDF0BC6316}"/>

<!--Sources-->
<!--Excel Source-->
<xs:enumeration value="{8C084929-27D1-479F-9641-ABB7CDADF1AC}"/>
<!--Flat File Source-->
<xs:enumeration value="{D23FD76B-F51D-420F-BBCB-19CBF6AC1AB4}"/>
<!--OLE DB Source-->
<xs:enumeration value="{165A526D-D5DE-47FF-96A6-F8274C19826B}"/>
<!--Raw File Source-->
<xs:enumeration value="{480C7D5A-CE63-405C-B338-3C7F26560EE3}"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="PipelineComponentPropertyNameEnum">
  <xs:restriction base="xs:string">
    <!--Transforms-->
    <!--Aggregate Task-->
    <xs:enumeration value="KeyScale"/>
    <xs:enumeration value="Keys"/>
    <xs:enumeration value="CountDistinctScale"/>
    <xs:enumeration value="CountDistinctKeys"/>
    <xs:enumeration value="AutoExtendFactor"/>
    <!--Audit (none)-->
    <!--Cache Transform (none)-->
    <!--Character Map (none)-->
    <!--Conditional Split (none)-->
    <!--Copy Column (none)-->
    <!--Data Conversion (none)-->
    <!--Data Mining Query-->
    <xs:enumeration value="ObjectRef"/>
    <xs:enumeration value="QueryText"/>
    <xs:enumeration value="CatalogName"/>
    <!--Derived Column Query (none)-->
    <!--Export Column Query (none)-->
    <!--Fuzzy Grouping-->
```

```xml
            <xs:enumeration value="MinSimilarity"/>
            <xs:enumeration value="Delimiters"/>
            <xs:enumeration value="MaxMemoryUsage"/>
            <xs:enumeration value="Exhaustive"/>
            <!--Fuzzy Lookup-->
            <xs:enumeration value="MatchIndexOptions"/>
            <xs:enumeration value="MaxMemoryUsage"/>
            <xs:enumeration value="MatchIndexName"/>
            <xs:enumeration value="ReferenceTableName"/>
            <xs:enumeration value="DropExistingMatchIndex"/>
            <xs:enumeration value="CopyReferenceTable"/>
            <xs:enumeration value="MaxOutputMatchesPerInput"/>
            <xs:enumeration value="MinSimilarity"/>
            <xs:enumeration value="Delimiters"/>
            <xs:enumeration value="ReferenceMetadataXml"/>
            <xs:enumeration value="Exhaustive"/>
            <xs:enumeration value="WarmCaches"/>
            <!--Import Column (none)-->
            <!--Lookup-->
            <xs:enumeration value="SqlCommand"/>
            <xs:enumeration value="SqlCommandParam"/>
            <xs:enumeration value="ConnectionType"/>
            <xs:enumeration value="CacheType"/>
            <xs:enumeration value="NoMatchBehavior"/>
            <xs:enumeration value="NoMatchCachePercentage"/>
            <xs:enumeration value="MaxMemoryUsage"/>
            <xs:enumeration value="MaxMemoryUsage64"/>
            <xs:enumeration value="ReferenceMetadataXml"/>
            <xs:enumeration value="ParameterMap"/>
            <xs:enumeration value="DefaultCodePage"/>
            <!--Merge (none)-->
            <!--MergeJoin (none)-->
            <xs:enumeration value="JoinType"/>
            <xs:enumeration value="NumKeyColumns"/>
            <xs:enumeration value="TreatNullsAsEqual"/>
            <xs:enumeration value="MaxBuffersPerInput"/>
            <!--Multicast (none)-->
            <!--OLE DB Command-->
            <xs:enumeration value="CommandTimeout"/>
            <xs:enumeration value="SqlCommand"/>
            <xs:enumeration value="DefaultCodePage"/>
            <!--Percentage Sampling-->
            <xs:enumeration value="SamplingValue"/>
            <xs:enumeration value="SamplingSeed"/>
            <!--Pivot (none)-->
            <!--RowCount-->
            <xs:enumeration value="VariableName"/>
            <!--Row Sampling-->
            <xs:enumeration value="SamplingValue"/>
            <xs:enumeration value="SamplingSeed"/>
            <!--Script Component-->
            <xs:enumeration value="SourceCode"/>
            <xs:enumeration value="BinaryCode"/>
            <xs:enumeration value="VSTAProjectName"/>
            <xs:enumeration value="ScriptLanguage"/>
            <xs:enumeration value="ReadOnlyVariables"/>
            <xs:enumeration value="ReadWriteVariables"/>
            <xs:enumeration value="BreakpointCollection"/>
            <xs:enumeration value="UserComponentTypeName"/>
            <!--Slowly Changing Dimension-->
            <xs:enumeration value="SqlCommand"/>
            <xs:enumeration value="CurrentRowWhere"/>
            <xs:enumeration value="UpdateChangingAttributeHistory"/>
            <xs:enumeration value="FailOnFixedAttributeChange"/>
            <xs:enumeration value="InferredMemberIndicator"/>
            <xs:enumeration value="EnableInferredMember"/>
            <xs:enumeration value="FailOnLookupFailure"/>
            <xs:enumeration value="IncomingRowChangeType"/>
            <xs:enumeration value="DefaultCodePage"/>
            <!--Sort-->
```

```
            <xs:enumeration value="EliminateDuplicates"/>
            <xs:enumeration value="MaximumThreads"/>
            <!--Term Extraction-->
            <xs:enumeration value="NeedReferenceData"/>
            <xs:enumeration value="OutTermTable"/>
            <xs:enumeration value="OutTermColumn"/>
            <xs:enumeration value="WordOrPhrase"/>
            <xs:enumeration value="ScoreType"/>
            <xs:enumeration value="FrequencyThreshold"/>
            <xs:enumeration value="MaxLengthOfTerm"/>
            <xs:enumeration value="IsCaseSensitive"/>
            <!--Term Lookup-->
            <xs:enumeration value="RefTermTable"/>
            <xs:enumeration value="RefTermColumn"/>
            <xs:enumeration value="IsCaseSensitive"/>
            <!--UnionAll (none)-->
            <!--Unpivot (none)-->

            <!--Data Destinations-->
            <!--ADO.Net Destination-->
            <xs:enumeration value="TableOrViewName"/>
            <xs:enumeration value="BatchSize"/>
            <xs:enumeration value="CommandTimeout"/>
            <xs:enumeration value="UserComponentTypeName"/>
            <!--Excel Destination-->
            <xs:enumeration value="CommandTimeout"/>
            <xs:enumeration value="OpenRowset"/>
            <xs:enumeration value="OpenRowsetVariable"/>
            <xs:enumeration value="SqlCommand"/>
            <xs:enumeration value="AccessMode"/>
            <xs:enumeration value="FastLoadKeepIdentity"/>
            <xs:enumeration value="FastLoadKeepNulls"/>
            <xs:enumeration value="FastLoadOptions"/>
            <xs:enumeration value="FastLoadMaxInsertCommitSize"/>
            <!--Flat File Destination-->
            <xs:enumeration value="Overwrite"/>
            <xs:enumeration value="Header"/>
            <!--OLE DB Destination-->
            <xs:enumeration value="CommandTimeout"/>
            <xs:enumeration value="OpenRowset"/>
            <xs:enumeration value="OpenRowsetVariable"/>
            <xs:enumeration value="SqlCommand"/>
            <xs:enumeration value="DefaultCodePage"/>
            <xs:enumeration value="AlwaysUseDefaultCodePage"/>
            <xs:enumeration value="AccessMode"/>
            <xs:enumeration value="FastLoadKeepIdentity"/>
            <xs:enumeration value="FastLoadKeepNulls"/>
            <xs:enumeration value="FastLoadOptions"/>
            <xs:enumeration value="FastLoadMaxInsertCommitSize"/>
            <!--Raw File Destination-->
            <xs:enumeration value="AccessMode"/>
            <xs:enumeration value="FileName"/>
            <xs:enumeration value="FileNameVariable"/>
            <xs:enumeration value="WriteOption"/>
            <!--DataReader Destination-->
            <xs:enumeration value="ReadTimeout"/>
            <xs:enumeration value="DataReader"/>
            <xs:enumeration value="FailOnTimeout"/>
            <xs:enumeration value="UserComponentTypeName"/>
            <!--Data Mining Model Training-->
            <xs:enumeration value="ObjectRef"/>
            <!--Dimension Processing-->
            <xs:enumeration value="ObjectRef"/>
            <xs:enumeration value="UseDefaultConfiguration"/>
            <xs:enumeration value="KeyErrorLimit"/>
            <xs:enumeration value="KeyErrorLogFile"/>
            <xs:enumeration value="KeyErrorAction"/>
            <xs:enumeration value="KeyErrorLimitAction"/>
            <xs:enumeration value="KeyNotFound"/>
            <xs:enumeration value="KeyDuplicate"/>
```

```xml
            <xs:enumeration value="NullKeyConvertedToUnknown"/>
            <xs:enumeration value="NullKeyNotAllowed"/>
            <xs:enumeration value="ProcessType"/>
            <!--Partition Processing-->
            <xs:enumeration value="ObjectRef"/>
            <xs:enumeration value="UseDefaultConfiguration"/>
            <xs:enumeration value="KeyErrorLimit"/>
            <xs:enumeration value="KeyErrorLogFile"/>
            <xs:enumeration value="KeyErrorAction"/>
            <xs:enumeration value="KeyErrorLimitAction"/>
            <xs:enumeration value="KeyNotFound"/>
            <xs:enumeration value="KeyDuplicate"/>
            <xs:enumeration value="NullKeyConvertedToUnknown"/>
            <xs:enumeration value="NullKeyNotAllowed"/>
            <xs:enumeration value="ProcessType"/>
            <!--RecordSet Destination-->
            <xs:enumeration value="VariableName"/>
            <!--SQL Server Compact Destination-->
            <xs:enumeration value="Table Name"/>
            <xs:enumeration value="UserComponentTypeName"/>
            <!--SQL Server Destination-->
            <xs:enumeration value="DefaultCodePage"/>
            <xs:enumeration value="AlwaysUseDefaultCodePage"/>
            <xs:enumeration value="BulkInsertTableName"/>
            <xs:enumeration value="BulkInsertCheckConstraints"/>
            <xs:enumeration value="BulkInsertFirstRow"/>
            <xs:enumeration value="BulkInsertFireTriggers"/>
            <xs:enumeration value="BulkInsertKeepIdentity"/>
            <xs:enumeration value="BulkInsertKeepNulls"/>
            <xs:enumeration value="BulkInsertLastRow"/>
            <xs:enumeration value="BulkInsertMaxErrors"/>
            <xs:enumeration value="BulkInsertOrder"/>
            <xs:enumeration value="BulkInsertTablock"/>
            <xs:enumeration value="Timeout"/>
            <xs:enumeration value="MaxInsertCommitSize"/>

            <!--Data Sources-->
            <!--ADO.Net Source-->
            <xs:enumeration value="SqlCommand"/>
            <xs:enumeration value="CommandTimeout"/>
            <xs:enumeration value="AllowImplicitStringConversion"/>
            <xs:enumeration value="TableOrViewName"/>
            <xs:enumeration value="AccessMode"/>
            <xs:enumeration value="UserComponentTypeName"/>
            <!--Excel Source-->
            <xs:enumeration value="CommandTimeout"/>
            <xs:enumeration value="OpenRowset"/>
            <xs:enumeration value="OpenRowsetVariable"/>
            <xs:enumeration value="SqlCommand"/>
            <xs:enumeration value="SqlCommandVariable"/>
            <xs:enumeration value="AccessMode"/>
            <xs:enumeration value="ParameterMapping"/>
            <!--Flat File Source-->
            <xs:enumeration value="RetainNulls"/>
            <xs:enumeration value="FileNameColumnName"/>
            <!--OLEDB Source Task-->
            <xs:enumeration value="CommandTimeout"/>
            <xs:enumeration value="OpenRowset"/>
            <xs:enumeration value="OpenRowsetVariable"/>
            <xs:enumeration value="SqlCommand"/>
            <xs:enumeration value="SqlCommandVariable"/>
            <xs:enumeration value="DefaultCodePage"/>
            <xs:enumeration value="AlwaysUseDefaultCodePage"/>
            <xs:enumeration value="AccessMode"/>
            <xs:enumeration value="ParameterMapping"/>
            <!--Raw File Source-->
            <xs:enumeration value="AccessMode"/>
            <xs:enumeration value="FileName"/>
            <xs:enumeration value="FileNameVariable"/>
            <!--XML Source-->
```

```
        <xs:enumeration value="UserComponentTypeName"/>
        <xs:enumeration value="XMLData"/>
        <xs:enumeration value="XMLDataVariable"/>
        <xs:enumeration value="AccessMode"/>
        <xs:enumeration value="XMLSchemaDefinition"/>
        <xs:enumeration value="UseInlineSchema"/>

    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="PipelineComponentOutputPropertyNameEnum">
  <xs:restriction base="xs:string">
    <!--Transforms-->
    <!--Aggregate Task-->
    <xs:enumeration value="KeyScale"/>
    <xs:enumeration value="Keys"/>
    <!--Audit (none)-->
    <!--Cache Transform (none)-->
    <!--Character Map (none)-->
    <!--Conditional Split-->
    <xs:enumeration value="Expression"/>
    <xs:enumeration value="FriendlyExpression"/>
    <xs:enumeration value="EvaluationOrder"/>
    <xs:enumeration value="IsDefaultOut"/>
    <!--Copy Column (none)-->
    <!--Data Conversion (none)-->
    <!--Data Mining Query(none)-->
    <!--Derived Column Query (none)-->
    <!--Export Column Query (none)-->
    <!--Fuzzy Grouping(none)-->
    <!--Fuzzy Lookup(none)-->
    <!--Import Column (none)-->
    <!--Lookup(none)-->
    <!--Merge (none)-->
    <!--MergeJoin (none)-->
    <!--Multicast (none)-->
    <!--OLE DB Command(none)-->
    <!--Percentage Sampling-->
    <xs:enumeration value="Selected"/>
    <!--Pivot (none)-->
    <!--RowCount (none)-->
    <!--Row Sampling-->
    <xs:enumeration value="Selected"/>
    <!--Script Component(none)-->
    <!--Slowly Changing Dimension(none)-->
    <!--Sort(none)-->
    <!--Term Extraction(none)-->
    <!--Term Lookup(none)-->
    <!--UnionAll (none)-->
    <!--Unpivot (none)-->

    <!--Data Sources(none)-->
    <!--ADO.Net Source(none)-->
    <!--Excel Source(none)-->
    <!--Flat File Source(none)-->
    <!--OLEDB Source Task(none)-->
    <!--Raw File Source(none)-->
    <!--XML Source-->
    <xs:enumeration value="RowsetID"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="PipelineComponentInputPropertyNameEnum">
  <xs:restriction base="xs:string">
  </xs:restriction>
  <!--None of the components have defined properties for
  the Input element-->
</xs:simpleType>

<xs:simpleType name="PipelineComponentOutputColumnPropertyNameEnum">
```

```xml
<xs:restriction base="xs:string">
  <!--Transforms-->
  <!--Aggregate Task-->
  <xs:enumeration value="AggregationColumnId"/>
  <xs:enumeration value="AggregationType"/>
  <xs:enumeration value="IsBig"/>
  <xs:enumeration value="AggregationComparisonFlags"/>
  <xs:enumeration value="CountDistinctKey"/>
  <xs:enumeration value="CountDistinctScale"/>
  <!--Audit-->
  <xs:enumeration value="LineageItemSelected"/>
  <!--Cache Transform (none)-->
  <!--Character Map (none)-->
  <!--Conditional Split (none)-->
  <!--Copy Column (none)-->
  <xs:enumeration value="copyColumnId"/>
  <!--Data Conversion (none)-->
  <xs:enumeration value="SourceInputColumnLineageID"/>
  <xs:enumeration value="FastParse"/>
  <!--Data Mining Query-->
  <!--Derived Column Query (none)-->
  <xs:enumeration value="Expression"/>
  <xs:enumeration value="FriendlyExpression"/>
  <!--Export Column Query (none)-->
  <!--Fuzzy Grouping-->
  <xs:enumeration value="ColumnType"/>
  <xs:enumeration value="InputId"/>
  <!--Fuzzy Lookup-->
  <xs:enumeration value="CopyFromReferenceColumn"/>
  <xs:enumeration value="SourceInputColumnLineageID"/>
  <xs:enumeration value="ColumnType"/>
  <!--Import Column (none)-->
  <!--Lookup-->
  <xs:enumeration value="CopyFromReferenceColumn"/>
  <!--Merge (none)-->
  <!--MergeJoin-->
  <xs:enumeration value="InputColumnID"/>
  <!--Multicast (none)-->
  <!--OLE DB Command (none)-->
  <!--Percentage Sampling (none)-->
  <!--Pivot-->
  <xs:enumeration value="SourceColumn"/>
  <xs:enumeration value="PivotKeyValue"/>
  <!--RowCount (none)-->
  <!--Row Sampling-->
  <xs:enumeration value="InputColumnLineageID"/>
  <!--Script Component (none)-->
  <!--Slowly Changing Dimension (none)-->
  <!--Sort-->
  <xs:enumeration value="SortColumnId"/>
  <!--Term Extraction(none)-->
  <!--Term Lookup-->
  <xs:enumeration value="CustomLineageID"/>
  <!--UnionAll (none)-->
  <!--Unpivot (none)-->
  <xs:enumeration value="PivotKey"/>

  <!--Data Sources-->
  <!--ADO.Net Source(none)-->
  <!--Excel Source(none)-->
  <!--Flat File Source-->
  <xs:enumeration value="FastParse"/>
  <xs:enumeration value="UseBinaryFormat"/>
  <!--OLEDB Source Task(none)-->
  <!--Raw File Source(none)-->
  <!--XML Source(none)-->
</xs:restriction>
</xs:simpleType>

<xs:simpleType name="PipelineComponentInputColumnPropertyNameEnum">
```

```xml
<xs:restriction base="xs:string">
  <!--Transforms-->
  <!--Aggregate Task-->
  <!--Audit (none)-->
  <!--Cache Transform-->
  <xs:enumeration value="CacheColumnName"/>
  <!--Character Map-->
  <xs:enumeration value="MapFlags"/>
  <!--Conditional Split (none)-->
  <!--Copy Column (none)-->
  <!--Data Conversion (none)-->
  <!--Data Mining Query(none)-->
  <!--Derived Column Query-->
  <xs:enumeration value="Expression"/>
  <xs:enumeration value="FriendlyExpression"/>
  <!--Export Column-->
  <xs:enumeration value="FileDataColumnID"/>
  <xs:enumeration value="WriteBOM"/>
  <xs:enumeration value="ForceTruncate"/>
  <xs:enumeration value="AllowAppend"/>
  <!--Fuzzy Grouping-->
  <xs:enumeration value="ToBeCleaned"/>
  <xs:enumeration value="ExactFuzzy"/>
  <xs:enumeration value="LeadingTrailingNumeralsSignificant"/>
  <xs:enumeration value="MinSimilarity"/>
  <xs:enumeration value="FuzzyComparisonFlags"/>
  <!--Fuzzy Lookup-->
  <xs:enumeration value="JoinType"/>
  <xs:enumeration value="ExactFuzzy"/>
  <xs:enumeration value="JoinToReferenceColumn"/>
  <xs:enumeration value="FuzzyComparisonFlags"/>
  <xs:enumeration value="FuzzyComparisonFlagsEx"/>
  <!--Import Column-->
  <xs:enumeration value="FileDataColumnID"/>
  <xs:enumeration value="ExpectBOM"/>
  <!--Lookup-->
  <xs:enumeration value="JoinToReferenceColumn"/>
  <xs:enumeration value="CopyFromReferenceColumn"/>
  <!--Merge-->
  <xs:enumeration value="OutputColumnLineageID"/>
  <!--MergeJoin (none)-->
  <!--Multicast (none)-->
  <!--OLE DB Command (none)-->
  <!--Percentage Sampling (none)-->
  <!--Pivot-->
  <xs:enumeration value="PivotUsage"/>
  <!--RowCount(none)-->
  <!--Row Sampling(none)-->
  <!--Script Component(none)-->
  <!--Slowly Changing Dimension-->
  <xs:enumeration value="ColumnType"/>
  <!--Sort-->
  <xs:enumeration value="NewComparisonFlags"/>
  <xs:enumeration value="NewSortKeyPosition"/>
  <!--Term Extraction(none)-->
  <!--Term Lookup-->
  <xs:enumeration value="InputColumnType"/>
  <!--UnionAll-->
  <xs:enumeration value="OutputColumnLineageID"/>
  <!--Unpivot-->
  <xs:enumeration value="DestinationColumn"/>
  <xs:enumeration value="PivotKeyValue"/>

  <!--Data Destinations-->
  <!--ADO.Net Destination(none)-->
  <!--Excel Destination(none)-->
  <!--Flat File Destination(none)-->
  <!--OLE DB Destination(none)-->
  <!--Raw File Destination(none)-->
  <!--DataReader Destination(none)-->
```

```xml
      <!--Data Mining Model Training(none)-->
      <!--Dimension Processing(none)-->
      <!--Partition Processing(none)-->
      <!--RecordSet Destination(none)-->
      <!--SQL Server Compact Destination(none)-->
      <!--SQL Server Destination(none)-->
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="PipelineComponentDataTypeEnum">
    <xs:restriction base="xs:string">
      <xs:enumeration value="System.Boolean"/>
      <xs:enumeration value="System.Byte"/>
      <xs:enumeration value="System.SByte"/>
      <xs:enumeration value="System.Decimal"/>
      <xs:enumeration value="System.Double"/>
      <xs:enumeration value="System.Single"/>
      <xs:enumeration value="System.Int32"/>
      <xs:enumeration value="System.UInt32"/>
      <xs:enumeration value="System.Int64"/>
      <xs:enumeration value="System.UInt64"/>
      <xs:enumeration value="System.Int16"/>
      <xs:enumeration value="System.UInt16"/>
      <xs:enumeration value="System.String"/>
      <xs:enumeration value="System.Null"/>
      <xs:enumeration value="System.DateTime"/>
      <xs:enumeration value="System.Object"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="PipelineComponentStateEnum">
    <xs:restriction base="xs:string">
      <xs:enumeration value="cdata"/>
      <xs:enumeration value="escaped"/>
      <xs:enumeration value="default"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="PipelineComponentExpressionTypeEnum">
    <xs:restriction base="xs:string">
      <xs:enumeration value="None"/>
      <xs:enumeration value="Notify"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="PipelineComponentConnectionType">
    <xs:sequence/>
    <xs:attribute name="refId" type="xs:string" use="required"
                  form="unqualified"/>
    <xs:attribute name="name" type="xs:string" use="optional"
                  default="" form="unqualified"/>
    <xs:attribute name="description" type="xs:string"
                  use="optional" default="" form="unqualified"/>
    <xs:attribute name="connectionManagerID" type="xs:string"
                  use="optional" default="" form="unqualified"/>
  </xs:complexType>

  <xs:complexType name="PipelineComponentInputType">
    <xs:sequence>
      <xs:element name="properties"
                  type="DTS:PipelineComponentInputPropertiesType"
                  minOccurs="0" form="unqualified"/>
      <xs:element name="inputColumns"
                  type="DTS:PipelineComponentInputColumnsType"
                  minOccurs="0" form="unqualified"/>
      <xs:element name="externalMetadataColumns"
                  type="DTS:PipelineComponentMetadataColumnsType"
                  minOccurs="0" form="unqualified"/>
    </xs:sequence>
    <xs:attributeGroup
```

```xml
                ref="DTS:PipelineComponentInputOutputElementAttributeGroup"/>
        </xs:complexType>

        <xs:complexType name="PipelineComponentInputColumnsType">
          <xs:sequence>
            <xs:element name="inputColumn"
                        type="DTS:PipelineComponentInputColumnType"
                        minOccurs="0" maxOccurs="unbounded" form="unqualified"/>
          </xs:sequence>
        </xs:complexType>

        <xs:complexType name="PipelineComponentInputColumnType">
          <xs:sequence>
            <xs:element name="properties"
                        type="DTS:PipelineComponentInputColumnPropertiesType"
                        form="unqualified" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attributeGroup
            ref="DTS:PipelineComponentAllColumnBaseAttributeGroup"/>
          <xs:attributeGroup
            ref="DTS:PipelineComponentIOColumnBaseAttributeGroup"/>
          <xs:attribute name="usageType"  form="unqualified"
                        use="optional" default="readOnly"
                        type="DTS:PipelineComponentUsageTypeEnum"/>
          <xs:attribute name="cachedName" type="xs:string"
                        use="optional" default="" form="unqualified"/>
          <xs:attribute name="cachedSortKeyPosition" type="xs:int"
                        use="optional" default="0" form="unqualified"/>
          <xs:attribute name="cachedPrecision" type="xs:int" use="optional"
                        default="0" form="unqualified"/>
          <xs:attribute name="cachedScale" type="xs:int" use="optional"
                        default="0" form="unqualified"/>
          <xs:attribute name="cachedLength" type="xs:int" use="optional"
                        default="0" form="unqualified"/>
          <xs:attribute name="cachedDataType"
                        type="DTS:PipelineComponentColumnDataTypeEnum"
                        use="optional" default="empty" form="unqualified"/>
          <xs:attribute name="cachedCodepage" type="xs:int" use="optional"
                        default="0" form="unqualified"/>
          <xs:attribute name="cachedComparisonFlags" type="xs:int"
                        form="unqualified" use="optional" default="0" />
        </xs:complexType>

        <xs:complexType name="PipelineComponentInputColumnPropertiesType">
          <xs:sequence>
            <xs:element name="property" form="unqualified"
                        type="DTS:PipelineComponentInputColumnPropertyType"
                        minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>

        <xs:complexType name="PipelineComponentInputColumnPropertyType">
          <xs:simpleContent>
            <xs:extension base="xs:anySimpleType">
              <xs:attributeGroup
                ref="DTS:PipelineComponentPropertyAttributeGroup"/>
              <xs:attribute name="name" form="unqualified" use="required"
                   type="DTS:PipelineComponentInputColumnPropertyNameEnum"/>
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>

        <xs:complexType name="PipelineComponentOutputColumnPropertiesType">
          <xs:sequence>
            <xs:element name="property" form="unqualified"
                        type="DTS:PipelineComponentOutputColumnPropertyType"
                        minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
```

```
<xs:complexType name="PipelineComponentOutputColumnPropertyType">
  <xs:simpleContent>
    <xs:extension base="xs:anySimpleType">
      <xs:attributeGroup
        ref="DTS:PipelineComponentPropertyAttributeGroup"/>
      <xs:attribute name="name" form="unqualified" use="required"
          type="DTS:PipelineComponentOutputColumnPropertyNameEnum"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

<xs:simpleType name="PipelineComponentUsageTypeEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="readOnly"/>
    <xs:enumeration value="readWrite"/>
    <xs:enumeration value="Ignored"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="PipelineComponentOutputType">
  <xs:sequence>
    <xs:element name="properties"
                type="DTS:PipelineComponentOutputPropertiesType"
                minOccurs="0" form="unqualified"/>
    <xs:element name="outputColumns"
                type="DTS:PipelineComponentOutputColumnsType"
                minOccurs="0" form="unqualified"/>
    <xs:element name="externalMetadataColumns"
                type="DTS:PipelineComponentMetadataColumnsType"
                minOccurs="0" form="unqualified"/>
  </xs:sequence>
  <xs:attributeGroup
    ref="DTS:PipelineComponentInputOutputElementAttributeGroup"/>
  <xs:attribute name="exclusionGroup" type="xs:int"
                use="optional" default="0" form="unqualified"/>
  <xs:attribute name="synchronousInputId" type="xs:string"
                use="optional" default="0" form="unqualified"/>
  <xs:attribute name="deleteOutputOnPathDetached" type="xs:boolean"
                use="optional" default="false" form="unqualified"/>
  <xs:attribute name="isErrorOut" type="xs:boolean"
                use="optional" default="false" form="unqualified"/>
  <xs:attribute name="isSorted" type="xs:boolean" use="optional"
                default="false" form="unqualified"/>
</xs:complexType>

<xs:complexType name="PipelineComponentMetadataColumnsType">
  <xs:sequence>
    <xs:element name="externalMetadataColumn"
        type="DTS:PipelineComponentMetadataColumnType"
        minOccurs="0" maxOccurs="unbounded" form="unqualified"/>
  </xs:sequence>
  <xs:attribute name="isUsed" form="unqualified">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="True"/>
        <xs:enumeration value="False"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:complexType>

<xs:complexType name="PipelineComponentMetadataColumnType">
  <xs:sequence>
    <xs:element name="properties"
        type="DTS:PipelineComponentMetadataColumnPropertiesType"
                minOccurs="0" form="unqualified"/>
  </xs:sequence>
  <xs:attributeGroup
    ref="DTS:PipelineComponentAllColumnBaseAttributeGroup"/>
  <xs:attributeGroup
```

```
                    ref="DTS:PipelineComponentColumnExtendedAttributeGroup"/>
        </xs:complexType>

        <xs:complexType name="PipelineComponentMetadataColumnPropertiesType">
          <xs:sequence>
            <xs:element name="property"
                type="DTS:PipelineComponentMetadataColumnPropertyType"
                minOccurs="0" maxOccurs="unbounded" form="unqualified"/>
          </xs:sequence>
        </xs:complexType>

        <xs:complexType name="PipelineComponentMetadataColumnPropertyType">
          <xs:simpleContent>
            <xs:extension base="xs:anySimpleType">
              <xs:attributeGroup
                ref="DTS:PipelineComponentPropertyAttributeGroup"/>
              <xs:attribute name="name" form="unqualified" use="required"
                  type="DTS:PipelineComponentMetadataColumnPropertyNameEnum"/>
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>

        <xs:simpleType name="PipelineComponentMetadataColumnPropertyNameEnum">
          <xs:restriction base="xs:string">
            <!--OLE DB Command-->
            <xs:enumeration value="DBParamInfoFlags"/>
            <!--XML Source-->
            <xs:enumeration value="CLRType"/>
          </xs:restriction>
        </xs:simpleType>

        <xs:attributeGroup
          name="PipelineComponentInputOutputElementAttributeGroup">
          <xs:attribute name="refId" type="xs:string" use="required"
                        form="unqualified"/>
          <xs:attribute name="name" type="xs:string" use="optional" default=""
                        form="unqualified"/>
          <xs:attribute name="description" type="xs:string"
                        use="optional" default="" form="unqualified"/>
          <xs:attribute name="hasSideEffects" type="xs:boolean"
                        use="optional" default="false" form="unqualified"/>
          <xs:attribute name="dangling" type="xs:boolean"
                        use="optional" default="false" form="unqualified"/>
          <xs:attribute name="errorOrTruncationOperation" type="xs:string"
                        use="optional" default="" form="unqualified"/>
          <xs:attribute name="errorRowDisposition" form="unqualified"
                        type="DTS:PipelineComponentRowDispositionEnum"
                        use="optional" default="NotUsed"/>
          <xs:attribute name="truncationRowDisposition" form="unqualified"
                        type="DTS:PipelineComponentRowDispositionEnum"
                        use="optional" default="NotUsed"/>
        </xs:attributeGroup>

        <xs:attributeGroup name="PipelineComponentAllColumnBaseAttributeGroup">
<!--       <xs:attribute name="id" type="xs:string" form="unqualified"/>-->
          <xs:attribute name="refId" type="xs:string" use="required"
                        form="unqualified"/>
          <xs:attribute name="name" type="xs:string" use="optional" default=""
                        form="unqualified"/>
          <xs:attribute name="description" type="xs:string" use="optional"
                        default="" form="unqualified"/>
          <xs:attribute name="mappedColumnId" type="xs:int" use="optional"
                        default="0" form="unqualified"/>
        </xs:attributeGroup>

        <xs:attributeGroup name="PipelineComponentIOColumnBaseAttributeGroup">
          <xs:attribute name="errorOrTruncationOperation" type="xs:string"
                        use="optional" default="" form="unqualified"/>
          <xs:attribute name="errorRowDisposition"
                        type="DTS:PipelineComponentRowDispositionEnum"
```

```
                    form="unqualified" use="optional" default="NotUsed"/>
   <xs:attribute name="truncationRowDisposition"
                    type="DTS:PipelineComponentRowDispositionEnum"
                    form="unqualified" use="optional" default="NotUsed"/>
   <xs:attribute name="externalMetadataColumnId" type="xs:string"
                    form="unqualified"/>
   <xs:attribute name="lineageId" type="xs:string" use="required"
                    form="unqualified"/>
</xs:attributeGroup>

<xs:attributeGroup name="PipelineComponentColumnExtendedAttributeGroup">
  <xs:attribute name="precision" type="xs:int" use="optional" default="0"
                    form="unqualified"/>
  <xs:attribute name="scale" type="xs:int" use="optional" default="0"
                    form="unqualified"/>
  <xs:attribute name="length" type="xs:int" use="optional" default="0"
                    form="unqualified"/>
  <xs:attribute name="dataType"
                    type="DTS:PipelineComponentColumnDataTypeEnum"
                    use="optional" default="empty" form="unqualified"/>
  <xs:attribute name="codePage" type="xs:int" use="optional" default="0"
                    form="unqualified"/>
</xs:attributeGroup>

<xs:complexType name="PipelineComponentOutputColumnsType">
  <xs:sequence>
    <xs:element name="outputColumn"
                    type="DTS:PipelineComponentOutputColumnType"
         minOccurs="0" maxOccurs="unbounded" form="unqualified"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="PipelineComponentOutputColumnType">
  <xs:sequence>
    <xs:element name="properties"
                    type="DTS:PipelineComponentOutputColumnPropertiesType"
                    form="unqualified" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attributeGroup
    ref="DTS:PipelineComponentAllColumnBaseAttributeGroup"/>
  <xs:attributeGroup
    ref="DTS:PipelineComponentIOColumnBaseAttributeGroup"/>
  <xs:attributeGroup
    ref="DTS:PipelineComponentColumnExtendedAttributeGroup"/>
  <xs:attribute name="sortKeyPosition" type="xs:int"
                    use="optional" default="0" form="unqualified"/>
  <xs:attribute name="comparisonFlags" type="xs:int"
                    form="unqualified" use="optional" default="0" />
  <xs:attribute name="specialFlags" form="unqualified"
                    use="optional" default="0">
    <xs:simpleType>
      <xs:restriction base="xs:int">
        <xs:minInclusive value="0"/>
        <xs:maxInclusive value="2"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:complexType>

<xs:simpleType name="PipelineComponentColumnDataTypeEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="empty"/>
    <xs:enumeration value="i1"/>
    <xs:enumeration value="ui1"/>
    <xs:enumeration value="i2"/>
    <xs:enumeration value="i4"/>
    <xs:enumeration value="r4"/>
    <xs:enumeration value="r8"/>
    <xs:enumeration value="cy"/>
    <xs:enumeration value="date"/>
```

```
          <xs:enumeration value="bool"/>
          <xs:enumeration value="variant"/>
          <xs:enumeration value="decimal"/>
          <xs:enumeration value="ui2"/>
          <xs:enumeration value="ui4"/>
          <xs:enumeration value="i8"/>
          <xs:enumeration value="ui8"/>
          <xs:enumeration value="guid"/>
          <xs:enumeration value="bytes"/>
          <xs:enumeration value="str"/>
          <xs:enumeration value="wstr"/>
          <xs:enumeration value="numeric"/>
          <xs:enumeration value="dbDate"/>
          <xs:enumeration value="dateTime"/>
          <xs:enumeration value="dbTime"/>
          <xs:enumeration value="dbTimeStamp"/>
          <xs:enumeration value="image"/>
          <xs:enumeration value="text"/>
          <xs:enumeration value="nText"/>
          <xs:enumeration value="dbTime2"/>
          <xs:enumeration value="dbTimeStamp2"/>
          <xs:enumeration value="dbTimeStampOffset"/>
      </xs:restriction>
   </xs:simpleType>

   <xs:simpleType name="PipelineComponentRowDispositionEnum">
      <xs:restriction base="xs:string">
         <xs:enumeration value="NotUsed"/>
         <xs:enumeration value="IgnoreFailure"/>
         <xs:enumeration value="FailComponent"/>
         <xs:enumeration value="RedirectRow"/>
      </xs:restriction>
   </xs:simpleType>

   <xs:complexType name="PipelinePathType">
      <xs:sequence/>
      <xs:attribute name="refId" type="xs:string" use="required"
                    form="unqualified"/>
      <xs:attribute name="id" type="xs:string" form="unqualified"/>
      <xs:attribute name="name" type="xs:string" form="unqualified"/>
      <xs:attribute name="description" type="xs:string"
                    form="unqualified"/>
      <xs:attribute name="startId" type="xs:string" form="unqualified"/>
      <xs:attribute name="endId" type="xs:string" form="unqualified"/>
   </xs:complexType>

   <!--ObjectData for non-Executables-->

   <xs:complexType name="LogProviderObjectDataType">
      <xs:sequence>
         <xs:element name="InnerObject" form="unqualified"
                     type="DTS:InnerObjectObjectDataType"/>
      </xs:sequence>
   </xs:complexType>

   <xs:complexType name="ForEachEnumeratorObjectDataType">
      <xs:choice>
         <xs:element name="FEIEItems" type="DTS:FEIEItemsType"
                     form="unqualified"/>
         <xs:element name="ForEachFileEnumeratorProperties"
                     type="DTS:ForEachFileEnumeratorPropertiesType"
                     form="unqualified"/>
         <xs:element name="FEEADO" type="DTS:FEEADOType"
                     form="unqualified"/>
         <xs:element name="FEESchemaRowset" type="DTS:FEESchemaRowsetType"
                     form="unqualified"/>
         <xs:element name="FEEFVE" type="DTS:FEEFVEType" form="unqualified"/>
         <xs:element name="FEENODELIST" type="DTS:FEENODELISTType"
                     form="unqualified"/>
         <xs:element name="FEESMO" type="DTS:FEESMOType" form="unqualified"/>
```

```
        </xs:choice>
    </xs:complexType>

    <xs:complexType name="FEIEItemsType">
      <xs:sequence>
        <xs:element name="FEIEItem" type="DTS:FEIEItemType"
                    maxOccurs="unbounded" form="unqualified"/>
      </xs:sequence>
    </xs:complexType>

    <xs:complexType name="FEIEItemType">
      <xs:sequence>
        <xs:element name="FEIEItemValue" maxOccurs="unbounded"
                    form="unqualified" type="DTS:FEIEItemValueType"/>
      </xs:sequence>
    </xs:complexType>

    <xs:complexType name="FEIEItemValueType">
      <xs:sequence/>
      <xs:attribute name="Type" type="DTS:DtsDataTypeEnum"
                    use="required" form="unqualified"/>
      <xs:attribute name="Value" type="xs:string"
                    use="required" form="unqualified"/>
    </xs:complexType>

    <xs:complexType name="ForEachFileEnumeratorPropertiesType">
      <xs:sequence>
        <xs:element name="FEFEProperty" maxOccurs="unbounded" form="unqualified">
          <xs:complexType>
            <xs:sequence/>
            <xs:attribute name="Folder" form="unqualified"/>
            <xs:attribute name="FileSpec" form="unqualified"/>
            <xs:attribute name="FileNameRetrievalType" form="unqualified">
              <xs:simpleType>
                <xs:restriction base="xs:int">
                  <xs:minInclusive value="0"/>
                  <xs:maxInclusive value="2"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="Recurse" form="unqualified">
              <xs:simpleType>
                <xs:restriction base="xs:int">
                  <xs:minInclusive value="-1"/>
                  <xs:maxInclusive value="0"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:attribute>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>

    <xs:complexType name="FEEADOType">
      <xs:sequence/>
      <xs:attribute name="EnumType" use="required" form="unqualified"
                    type="DTS:FEEADOEnumType"/>
      <xs:attribute name="VarName" type="xs:string"
                    use="required" form="unqualified"/>
    </xs:complexType>

    <xs:simpleType name="FEEADOEnumType">
      <xs:restriction base="xs:string">
        <xs:enumeration value="EnumerateRowsInFirstTable"/>
        <xs:enumeration value="EnumerateAllRows"/>
        <xs:enumeration value="EnumerateTables"/>
      </xs:restriction>
    </xs:simpleType>

    <xs:complexType name="FEESchemaRowsetType">
```

```xml
    <xs:sequence>
      <xs:element name="Restriction" maxOccurs="unbounded"
                  form="unqualified" type="DTS:RestrictionElementType"/>
    </xs:sequence>
    <xs:attribute name="Connection" type="xs:string"
                  use="required" form="unqualified"/>
    <xs:attribute name="Schema" type="DTS:uuid"
                  use="required" form="unqualified"/>
  </xs:complexType>

  <xs:complexType name="RestrictionElementType">
    <xs:sequence/>
    <xs:attribute name="Index" type="xs:int" use="required"
                  form="unqualified"/>
    <xs:attribute name="Value" type="xs:string" use="required"
                  form="unqualified"/>
  </xs:complexType>

  <xs:complexType name="FEEFVEType">
    <xs:sequence/>
    <xs:attribute name="VariableName" type="xs:string" form="unqualified" use="required"/>
  </xs:complexType>

  <xs:complexType name="FEENODELISTType">
    <xs:sequence/>
    <xs:attribute name="EnumerationType"
                  type="DTS:NodelistEnumerationTypeEnum"
                  form="unqualified" use="required"/>
    <xs:attribute name="OuterXPathSourceType" form="unqualified"
                  use="required" type="DTS:SourceTypeEnum"/>
    <xs:attribute name="OuterXPathString" type="xs:string"
                  form="unqualified" use="required"/>
    <xs:attribute name="InnerElementType"
                  type="DTS:NodelistInnerElementEnumerationTypeEnum"
                  use="required" form="unqualified"/>
    <xs:attribute name="InnerXPathSourceType" form="unqualified"
                  use="required" type="DTS:SourceTypeEnum"/>
    <xs:attribute name="InnerXPathString" type="xs:string"
                  form="unqualified" use="required"/>
    <xs:attribute name="SourceType" form="unqualified"
                  use="required" type="DTS:SourceTypeEnum"/>
    <xs:attribute name="SourceDocument" type="xs:string"
                  use="required" form="unqualified"/>
  </xs:complexType>

  <xs:simpleType name="NodelistInnerElementEnumerationTypeEnum">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Navigator"/>
      <xs:enumeration value="Node"/>
      <xs:enumeration value="NodeText"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="NodelistEnumerationTypeEnum">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Navigator"/>
      <xs:enumeration value="Node"/>
      <xs:enumeration value="NodeText"/>
      <xs:enumeration value="ElementCollection"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="FEESMOType">
    <xs:sequence/>
    <xs:attribute name="EnumURN" type="xs:string" form="unqualified" use="required"/>
  </xs:complexType>

  <xs:complexType name="ConnectionManagerObjectDataType">
    <xs:choice>
      <xs:element name="ConnectionManager" type=
```

```
            "DTS:ConnectionManagerObjectDataConnectionManagerType"/>
      <xs:element name="MsmqConnectionManager" type=
         "DTS:ConnectionManagerObjectDataMsmqConnectionManagerType"
                  form="unqualified"/>
      <xs:element name="SMOServerConnectionManager" type=
         "DTS:ConnectionManagerObjectDataSMOServerConnectionManagerType"
                  form="unqualified"/>
      <xs:element name="SmtpConnectionManager" type=
         "DTS:ConnectionManagerObjectDataSmtpConnectionManagerType"
                  form="unqualified"/>
      <xs:element name="WmiConnectionManager" type=
         "DTS:ConnectionManagerObjectDataWmiConnectionManagerType"
                  form="unqualified"/>
   </xs:choice>
</xs:complexType>

<xs:complexType
   name="ConnectionManagerObjectDataConnectionManagerType">
   <xs:sequence>
     <xs:choice minOccurs="0">
       <xs:element name="FlatFileColumns" type="DTS:FlatFileColumnsType"/>
       <xs:element name="CacheColumns" type="DTS:CacheColumnsType"
                  maxOccurs="unbounded"/>
       <xs:element name="FtpConnection" type="DTS:FtpConnectionType"/>
       <xs:element name="HttpConnection" type="DTS:HttpConnectionType"/>
     </xs:choice>
   </xs:sequence>
   <xs:attributeGroup
      ref="DTS:ConnectionManagerConnectionManagerAttributeGroup"/>
</xs:complexType>

<xs:complexType name="FlatFileColumnsType">
   <xs:sequence>
     <xs:element name="FlatFileColumn" type="DTS:FlatFileColumnType"
                 minOccurs="0" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>

<xs:complexType name="FlatFileColumnType">
   <xs:sequence/>
   <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
   <xs:attributeGroup ref="DTS:FlatFileColumnAttributeGroup"/>
</xs:complexType>

<xs:complexType name="CacheColumnsType">
   <xs:sequence>
     <xs:element name="CacheColumn" type="DTS:CacheColumnType"
                 minOccurs="0" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>

<xs:complexType name="CacheColumnType">
   <xs:sequence/>
   <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
   <xs:attributeGroup ref="DTS:CacheColumnAttributeGroup"/>
</xs:complexType>

<xs:complexType name="FtpConnectionType">
   <xs:sequence>
     <xs:element name="Property"
                 minOccurs="0" maxOccurs="unbounded">
       <xs:complexType>
         <xs:complexContent>
           <xs:extension base="DTS:PropertyElementBaseType">
             <xs:attribute name="Name"
                 type="DTS:FtpConnectionPropertyNameEnum"
                 use="required"/>
             <xs:attribute name="Sensitive" form="unqualified">
               <xs:simpleType>
                 <xs:restriction base="xs:int">
```

```xml
              <xs:minInclusive value="0"/>
              <xs:maxInclusive value="1"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="Encrypted" form="unqualified">
          <xs:simpleType>
            <xs:restriction base="xs:int">
              <xs:minInclusive value="0"/>
              <xs:maxInclusive value="1"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
    </xs:sequence>
    <xs:attributeGroup ref="DTS:FtpConnectionAttributeGroup"/>
</xs:complexType>

<xs:complexType name="HttpConnectionType">
  <xs:sequence>
    <xs:element name="Property"
                minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:complexContent>
          <xs:extension base="DTS:PropertyElementBaseType">
            <xs:attribute name="Name"
                type="DTS:HttpConnectionPropertyNameEnum"
                use="required"/>
            <xs:attribute name="Sensitive" form="unqualified">
              <xs:simpleType>
                <xs:restriction base="xs:int">
                  <xs:minInclusive value="0"/>
                  <xs:maxInclusive value="1"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="Encrypted" form="unqualified">
              <xs:simpleType>
                <xs:restriction base="xs:int">
                  <xs:minInclusive value="0"/>
                  <xs:maxInclusive value="1"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:attribute>
          </xs:extension>
        </xs:complexContent>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attributeGroup ref="DTS:HttpConnectionAttributeGroup"/>
</xs:complexType>

<xs:complexType
  name="ConnectionManagerObjectDataMsmqConnectionManagerType">
  <xs:sequence/>
  <xs:attribute name="ConnectionString" type="xs:string"
                form="unqualified"/>
</xs:complexType>

<xs:complexType
  name="ConnectionManagerObjectDataSMOServerConnectionManagerType">
  <xs:sequence>
    <xs:element name="PassWord" type=
                "DTS:PasswordElementType"
                form="unqualified"/>
  </xs:sequence>
  <xs:attribute name="ConnectionString" type="xs:string"
```

```
                       form="unqualified" use="required"/>
      </xs:complexType>

      <xs:complexType name="PasswordElementType">
        <xs:simpleContent>
          <xs:extension base="xs:string">
            <xs:attribute name="Sensitive" form="unqualified">
              <xs:simpleType>
                <xs:restriction base="xs:int">
                  <xs:minInclusive value="0"/>
                  <xs:maxInclusive value="1"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="Encrypted" form="unqualified">
              <xs:simpleType>
                <xs:restriction base="xs:int">
                  <xs:minInclusive value="0"/>
                  <xs:maxInclusive value="1"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:attribute>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>

      <xs:complexType
        name="ConnectionManagerObjectDataSmtpConnectionManagerType">
        <xs:sequence/>
        <xs:attribute name="ConnectionString" type="xs:string"
                      form="unqualified"/>
      </xs:complexType>

      <xs:complexType
        name="ConnectionManagerObjectDataWmiConnectionManagerType">
        <xs:sequence>
          <xs:element name="PassWord" minOccurs="0"
                      type="DTS:PasswordElementType"
                      form="unqualified"/>
        </xs:sequence>
        <xs:attribute name="ConnectionString" type="xs:string"
                      form="unqualified"/>
      </xs:complexType>

      <!--Property Element Types-->

      <xs:complexType name="PropertyElementBaseType">
        <xs:simpleContent>
          <xs:extension base="xs:anySimpleType">
            <xs:attribute name="DataType" type="DTS:DtsDataTypeEnum"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>

      <xs:complexType name="PropertyExpressionElementType">
        <xs:simpleContent>
          <xs:extension base="xs:anySimpleType">
            <xs:attribute name="Name" type="xs:string" use="required"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>

      <!--Enumeration Types-->

      <xs:simpleType name="BasePropertyNameEnum">
        <xs:restriction base="xs:string">
          <xs:enumeration value="Description"/>
          <xs:enumeration value="DTSID"/>
          <xs:enumeration value="CreationName"/>
          <xs:enumeration value="ObjectName"/>
```

```
      </xs:restriction>
    </xs:simpleType>

    <xs:attributeGroup name="BasePropertyAttributeGroup">
      <xs:attribute name="Description" type="xs:string" default=""
                    use="optional" form="qualified"/>
      <xs:attribute name="DTSID" type="DTS:uuid" use="required"
                    form="qualified"/>
      <xs:attribute name="CreationName" type="xs:string" use="required"
                    form="qualified"/>
      <xs:attribute name="ObjectName" type="xs:string" use="optional"
                    form="qualified"/>
    </xs:attributeGroup>

    <xs:simpleType name="DtsDataTypeEnum">
      <xs:restriction base="xs:int">
        <xs:enumeration value="0"/>
        <xs:enumeration value="1"/>
        <xs:enumeration value="2"/>
        <xs:enumeration value="3"/>
        <xs:enumeration value="4"/>
        <xs:enumeration value="5"/>
        <xs:enumeration value="6"/>
        <xs:enumeration value="7"/>
        <xs:enumeration value="8"/>
        <xs:enumeration value="11"/>
        <xs:enumeration value="13"/>
        <xs:enumeration value="14"/>
        <xs:enumeration value="16"/>
        <xs:enumeration value="17"/>
        <xs:enumeration value="18"/>
        <xs:enumeration value="19"/>
        <xs:enumeration value="20"/>
        <xs:enumeration value="21"/>
        <xs:enumeration value="22"/>
        <xs:enumeration value="23"/>
        <xs:enumeration value="64"/>
        <xs:enumeration value="72"/>
        <xs:enumeration value="129"/>
        <xs:enumeration value="130"/>
        <xs:enumeration value="131"/>
        <xs:enumeration value="133"/>
        <xs:enumeration value="134"/>
        <xs:enumeration value="135"/>
        <xs:enumeration value="139"/>
        <xs:enumeration value="145"/>
        <xs:enumeration value="146"/>
      </xs:restriction>
    </xs:simpleType>

    <xs:attributeGroup name="AllExecutableAttributeGroup">
      <xs:attribute name="DisableEventHandlers" type="DTS:BooleanStringCap"
                    default="False" use="optional" form="qualified"/>
    </xs:attributeGroup>

    <xs:simpleType name="BaseExecutablePropertyNameEnum">
      <xs:union memberTypes="DTS:BasePropertyNameEnum">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="ForceExecValue"/>
            <xs:enumeration value="ExecValue"/>
            <xs:enumeration value="ForceExecutionResult"/>
            <xs:enumeration value="Disabled"/>
            <xs:enumeration value="FailPackageOnFailure"/>
            <xs:enumeration value="FailParentOnFailure"/>
            <xs:enumeration value="MaxErrorCount"/>
            <xs:enumeration value="ISOLevel"/>
            <xs:enumeration value="LocaleID"/>
            <xs:enumeration value="TransactionOption"/>
            <xs:enumeration value="DelayValidation"/>
```

```
        </xs:restriction>
      </xs:simpleType>
    </xs:union>
  </xs:simpleType>

    <xs:attributeGroup name="BaseExecutablePropertyAttributeGroup">
      <xs:attribute name="ForceExecValue" type="DTS:BooleanStringCap"
                    default="False" use="optional" form="qualified"/>
      <xs:attribute name="ExecValue" type="xs:anySimpleType" default="0"
                    use="optional" form="qualified"/>
      <xs:attribute name="ForceExecutionResult" default="-1" use="optional"
                    type="DTS:ForceExecutionResultEnum" form="qualified"/>
      <xs:attribute name="Disabled" type="DTS:BooleanStringCap" default="False"
                    use="optional" form="qualified"/>
      <xs:attribute name="FailPackageOnFailure" type="DTS:BooleanStringCap"
                    default="False" use="optional" form="qualified"/>
      <xs:attribute name="FailParentOnFailure" type="DTS:BooleanStringCap"
                    default="False" use="optional" form="qualified"/>
      <xs:attribute name="MaxErrorCount" type="xs:int" default="1"
                    use="optional" form="qualified"/>
      <xs:attribute name="ISOLevel" type="DTS:ISOLevelEnum"
                    default="1048576" use="optional" form="qualified"/>
      <xs:attribute name="LocaleID" type="xs:int" use="optional"
                    form="qualified"/>
      <xs:attribute name="TransactionOption" type="DTS:TransactionOptionEnum"
                    default="1" use="optional" form="qualified"/>
      <xs:attribute name="DelayValidation" type="DTS:BooleanStringCap"
                    default="False" use="optional" form="qualified"/>
      <xs:attribute name="ExecValueType" type="DTS:DtsDataTypeEnum"
                    use="required" form="qualified"/>
    </xs:attributeGroup>
  <!--3 ENUM follow-->
  <xs:simpleType name="ForceExecutionResultEnum">
    <xs:restriction base="xs:int">
      <xs:minInclusive value="-1"/>
      <xs:maxInclusive value="2"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="ISOLevelEnum">
    <xs:restriction base="xs:int">
      <xs:enumeration value="-1"/>
      <xs:enumeration value="16"/>
      <xs:enumeration value="1033"/>
      <xs:enumeration value="4096"/>
      <xs:enumeration value="65536"/>
      <xs:enumeration value="1048576"/>
      <xs:enumeration value="16777216"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="TransactionOptionEnum">
    <xs:restriction base="xs:int">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="2"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="ExecutableTypePackagePropertyNameEnum">
    <xs:restriction base="xs:string">
      <xs:enumeration value="PackageFormatVersion"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:attributeGroup name="ExecutableTypePackageAttributeGroup">
    <xs:attribute name="VersionComments" type="xs:string" default=""
                  use="optional" form="qualified"/>
    <xs:attribute name="CreatorName" type="xs:string" use="optional"
                  form="qualified"/>
    <xs:attribute name="CreatorComputerName" type="xs:string"
```

```
                    use="optional" form="qualified"/>
    <xs:attribute name="CreationDate" type="xs:dateTime" use="optional"
                    form="qualified"/>
    <xs:attribute name="PackageType" type="DTS:PackageTypeEnum" default="0"
                    use="optional" form="qualified"/>
    <xs:attribute name="ProtectionLevel" type="DTS:ProtectionLevelEnum"
                    default="1" use="optional" form="qualified"/>
    <xs:attribute name="MaxConcurrentExecutables" type="xs:int"
                    default="-1" use="optional" form="qualified"/>
    <xs:attribute name="PackagePriorityClass" default="0" use="optional"
                    type="DTS:PackagePriorityClassEnum" form="qualified"/>
    <xs:attribute name="VersionMajor" type="xs:int" default="1"
                    use="optional" form="qualified"/>
    <xs:attribute name="VersionMinor" type="xs:int" default="0"
                    use="optional" form="qualified"/>
    <xs:attribute name="VersionBuild" type="xs:int" default="0"
                    use="optional" form="qualified"/>
    <xs:attribute name="VersionGUID" type="DTS:uuid" use="optional"
                    form="qualified"/>
    <xs:attribute name="EnableConfig" type="DTS:BooleanStringCap"
                    default="False" use="optional" form="qualified"/>
    <xs:attribute name="CheckpointFileName" type="xs:string" default=""
                    use="optional" form="qualified"/>
    <xs:attribute name="SaveCheckpoints" type="DTS:BooleanStringCap"
                    default="False" use="optional" form="qualified"/>
    <xs:attribute name="CheckpointUsage" type="DTS:CheckpointUsageEnum"
                    default="0" use="optional" form="qualified"/>
    <xs:attribute name="SuppressConfigurationWarnings" default="False"
                    type="DTS:BooleanStringCap" use="optional"
                    form="qualified"/>
    <xs:attribute name="LastModifiedProductVersion" type="xs:string"
                    use="optional" form="qualified"/>
</xs:attributeGroup>
<!--4 ENUM follow-->
<xs:simpleType name="PackageTypeEnum">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="6"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="ProtectionLevelEnum">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="5"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="PackagePriorityClassEnum">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="4"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="CheckpointUsageEnum">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="2"/>
  </xs:restriction>
</xs:simpleType>

<xs:attributeGroup name="AnyNonPackageExecutableAttributeGroup">
  <!--Pipeline & Tasks use these attributes-->
  <xs:attribute name="ExecutionLocation" type="xs:int" default="0"
                    use="optional" form="qualified"/>
  <xs:attribute name="ExecutionAddress" type="xs:string" default=""
                    use="optional" form="qualified"/>
  <xs:attribute name="TaskContact" type="xs:string" default=""
                    use="optional" form="qualified"/>
```

```
                <!--ForLoop uses these attributes-->
                <xs:attribute name="InitExpression" type="xs:string" default=""
                              use="optional" form="qualified"/>
                <xs:attribute name="EvalExpression" type="xs:string" use="optional"
                              form="qualified"/>
                <xs:attribute name="AssignExpression" type="xs:string" default=""
                              use="optional" form="qualified"/>
                <!--ForLoop and ForEachLoop use this attribute-->
                <xs:attribute name="MaxConcurrent" type="xs:int" default="1"
                              use="optional" form="qualified"/>
                <!--Sequence uses this attribute-->
                <xs:attribute name="LockName" type="xs:string" default=""
                              use="optional" form="qualified"/>
        </xs:attributeGroup>

        <xs:simpleType name="LogProviderPropertyNameEnum">
          <xs:union memberTypes="DTS:BasePropertyNameEnum">
            <xs:simpleType>
              <xs:restriction base="xs:string">
                <xs:enumeration value="DelayValidation"/>
                <xs:enumeration value="ConfigString"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:union>
        </xs:simpleType>

        <xs:attributeGroup name="LogProviderAttributeGroup">
          <xs:attribute name="DelayValidation" type="DTS:BooleanStringCap"
                        default="False" use="optional" form="qualified"/>
          <xs:attribute name="ConfigString" type="xs:string" default=""
                        use="optional" form="qualified"/>
        </xs:attributeGroup>

        <xs:simpleType name="PackageVariablePropertyNameEnum">
          <xs:union memberTypes="DTS:BasePropertyNameEnum">
            <xs:simpleType>
              <xs:restriction base="xs:string">
                <xs:enumeration value="PackageVariableValue"/>
                <xs:enumeration value="PackageVariable"/>
                <xs:enumeration value="Namespace"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:union>
        </xs:simpleType>

        <xs:simpleType name="ConfigurationPropertyNameEnum">
          <xs:union memberTypes="DTS:BasePropertyNameEnum">
            <xs:simpleType>
              <xs:restriction base="xs:string">
                <xs:enumeration value="ConfigurationType"/>
                <xs:enumeration value="ConfigurationString"/>
                <xs:enumeration value="ConfigurationVariable"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:union>
        </xs:simpleType>

        <xs:attributeGroup name="ConfigurationAttributeGroup">
          <xs:attribute name="ConfigurationType" type="DTS:ConfigurationTypeEnum"
                        default="0" use="optional" form="qualified"/>
          <xs:attribute name="ConfigurationString" type="xs:string" default=""
                        use="optional" form="qualified"/>
          <xs:attribute name="ConfigurationVariable" type="xs:string" default=""
                        use="optional" form="qualified"/>
        </xs:attributeGroup>
        <!--1 ENUM follows-->
        <xs:simpleType name="ConfigurationTypeEnum">
          <xs:restriction base="xs:int">
            <xs:minInclusive value="0"/>
            <xs:maxInclusive value="10"/>
```

```
      </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="PrecedenceConstraintPropertyNameEnum">
      <xs:union memberTypes="DTS:BasePropertyNameEnum">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="Value"/>
            <xs:enumeration value="EvalOp"/>
            <xs:enumeration value="LogicalAnd"/>
            <xs:enumeration value="Expression"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:union>
    </xs:simpleType>

    <xs:attributeGroup name="PrecedenceConstraintAttributeGroup">
      <xs:attribute name="Value" type="DTS:PrecedenceConstraintValueEnum"
                    use="optional" default="0" form="qualified"/>
      <xs:attribute name="EvalOp" type="DTS:PrecedenceConstraintEvalOpEnum"
                    use="optional" default="1" form="qualified"/>
      <xs:attribute name="LogicalAnd" type="DTS:BooleanStringCap"
                    use="optional" default="False" form="qualified"/>
      <xs:attribute name="Expression" type="xs:string" use="optional"
                    default="" form="qualified"/>
      <xs:attribute name="From" type="xs:string" use="required"
                    form="qualified"/>
      <xs:attribute name="To" type="xs:string" use="required"
                    form="qualified"/>
    </xs:attributeGroup>
    <!--2 ENUM follow-->
    <xs:simpleType name="PrecedenceConstraintValueEnum">
      <xs:restriction base="xs:int">
        <xs:minInclusive value="0"/>
        <xs:maxInclusive value="2"/>
      </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="PrecedenceConstraintEvalOpEnum">
      <xs:restriction base="xs:int">
        <xs:minInclusive value="1"/>
        <xs:maxInclusive value="4"/>
      </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="LoggingOptionsPropertyNameEnum">
      <xs:restriction base="xs:string">
        <xs:enumeration value="EventFilter"/>
        <xs:enumeration value="ColumnFilter"/>
      </xs:restriction>
    </xs:simpleType>

    <xs:attributeGroup name="LoggingOptionsAttributeGroup">
      <xs:attribute name="LoggingMode" type="DTS:LoggingModeEnum" default="0"
                    use="optional" form="qualified"/>
      <xs:attribute name="FilterKind" type="DTS:FilterKindEnum" default="1"
                    use="optional" form="qualified"/>
      <xs:attribute name="EventFilter" type="xs:string" default=""
                    use="optional" form="qualified"/>
    </xs:attributeGroup>
    <!--2 ENUM follow-->
    <xs:simpleType name="LoggingModeEnum">
      <xs:restriction base="xs:int">
        <xs:minInclusive value="0"/>
        <xs:maxInclusive value="2"/>
      </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="FilterKindEnum">
      <xs:restriction base="xs:int">
```

```
      <xs:enumeration value="0"/>
      <xs:enumeration value="1"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="ColumnEventFilterPropertyNameEnum">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Computer"/>
      <xs:enumeration value="Operator"/>
      <xs:enumeration value="SourceName"/>
      <xs:enumeration value="SourceID"/>
      <xs:enumeration value="ExecutionID"/>
      <xs:enumeration value="MessageText"/>
      <xs:enumeration value="DataBytes"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="ForEachVariableMappingPropertyNameEnum">
    <xs:union memberTypes="DTS:BasePropertyNameEnum">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="VariableName"/>
          <xs:enumeration value="ValueIndex"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:union>
  </xs:simpleType>

  <xs:attributeGroup name="ForEachVariableMappingAttributeGroup">
    <xs:attribute name="VariableName" type="xs:string" use="required"
                  form="qualified"/>
    <xs:attribute name="ValueIndex" type="xs:int" use="required"
                  form="qualified"/>
  </xs:attributeGroup>

  <xs:simpleType name="VariablePropertyNameEnum">
    <xs:union memberTypes="DTS:BasePropertyNameEnum">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="Namespace"/>
          <xs:enumeration value="Expression"/>
          <xs:enumeration value="EvaluateAsExpression"/>
          <xs:enumeration value="ReadOnly"/>
          <xs:enumeration value="RaiseChangedEvent"/>
          <xs:enumeration value="IncludeInDebugDump"/>
          <xs:enumeration value="Expression"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:union>
  </xs:simpleType>

  <xs:attributeGroup name="VariableAttributeGroup">
    <xs:attribute name="Namespace" type="xs:string" use="optional"
                  default="" form="qualified"/>
    <xs:attribute name="Expression" type="xs:string" use="optional"
                  default="" form="qualified"/>
    <xs:attribute name="EvaluateAsExpression" type="DTS:BooleanStringCap"
                  use="optional" default="False" form="qualified"/>
    <xs:attribute name="ReadOnly" type="DTS:BooleanStringCap" use="optional"
                  default="False" form="qualified"/>
    <xs:attribute name="RaiseChangedEvent" type="DTS:BooleanStringCap"
                  use="optional" default="False" form="qualified"/>
    <xs:attribute name="IncludeInDebugDump" type="xs:int" use="optional"
                  default="0" form="qualified"/>
  </xs:attributeGroup>
<!--1 ENUM follows-->
  <xs:simpleType name="IncludeInDebugDumpEnum">
    <xs:restriction base="xs:int">
      <xs:enumeration value="2345"/>
      <xs:enumeration value="6789"/>
```

```
        </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="BooleanStringCap">
      <xs:restriction base="xs:string">
        <xs:enumeration value="True"/>
        <xs:enumeration value="False"/>
      </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="ConnectionManagerPropertyNameEnum">
      <xs:union memberTypes="DTS:BasePropertyNameEnum">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="DelayValidation"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:union>
    </xs:simpleType>

    <xs:attributeGroup name="ConnectionManagerAttributeGroup">
      <xs:attribute name="DelayValidation" type="DTS:BooleanStringCap"
                    default="False" use="optional" form="qualified"/>
    </xs:attributeGroup>

    <xs:simpleType name="ConnectionManagerObjectDataPropertyNameEnum">
      <xs:restriction base="xs:string">
        <xs:enumeration value="Retain"/>
        <xs:enumeration value="ConnectionString"/>
        <xs:enumeration value="FileUsageType"/>
        <xs:enumeration value="Format"/>
        <xs:enumeration value="LocaleID"/>
        <xs:enumeration value="Unicode"/>
        <xs:enumeration value="HeaderRowsToSkip"/>
        <xs:enumeration value="HeaderRowDelimiter"/>
        <xs:enumeration value="ColumnNamesInFirstDataRow"/>
        <xs:enumeration value="RowDelimiter"/>
        <xs:enumeration value="DataRowsToSkip"/>
        <xs:enumeration value="TextQualifier"/>
        <xs:enumeration value="CodePage"/>
        <xs:enumeration value="ServerName"/>
        <xs:enumeration value="UseFile"/>
        <xs:enumeration value="UseEncryption"/>
        <xs:enumeration value="RetainData"/>
      </xs:restriction>
    </xs:simpleType>

    <xs:attributeGroup name="ConnectionManagerConnectionManagerAttributeGroup">
      <xs:attribute name="Retain" type="DTS:BooleanStringCap" default="False"
                    use="optional" form="qualified"/>
      <xs:attribute name="ConnectionString" type="xs:string" use="required"
                    form="qualified"/>
      <xs:attribute name="FileUsageType" type="DTS:FileUsageTypeEnum"
                    default="0" use="optional" form="qualified"/>
      <xs:attribute name="Format" type="DTS:FormatEnum" use="optional"
                    form="qualified"/>
      <xs:attribute name="LocaleID" type="xs:int" use="optional"
                    form="qualified"/>
      <xs:attribute name="Unicode" type="DTS:BooleanStringCap" default="False"
                    use="optional" form="qualified"/>
      <xs:attribute name="HeaderRowsToSkip" type="xs:int" default="0"
                    use="optional" form="qualified"/>
      <xs:attribute name="HeaderRowDelimiter" type="xs:string" use="optional"
                    form="qualified"/>
      <xs:attribute name="ColumnNamesInFirstDataRow" default="False"
                    type="DTS:BooleanStringCap" use="optional" form="qualified"/>
      <xs:attribute name="RowDelimiter" type="xs:string" use="optional"
                    form="qualified"/>
      <xs:attribute name="DataRowsToSkip" type="xs:int" default="0"
                    use="optional" form="qualified"/>
```

```
       <xs:attribute name="TextQualifier" type="xs:string" use="optional"
                     form="qualified"/>
       <xs:attribute name="CodePage" type="xs:int" use="optional"
                     form="qualified"/>
       <xs:attribute name="ServerName" type="xs:string" use="optional"
                     form="qualified"/>
       <xs:attribute name="UseFile" type="DTS:BooleanStringCap"
                     default="False" use="optional" form="qualified"/>
       <xs:attribute name="UseEncryption" type="DTS:BooleanStringCap"
                     default="False" use="optional" form="qualified"/>
       <xs:attribute name="RetainData" type="DTS:BooleanStringCap"
                     default="True" use="optional" form="qualified"/>
  </xs:attributeGroup>
  <!--2 ENUM follow-->
  <xs:simpleType name="FileUsageTypeEnum">
    <xs:restriction base="xs:int">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="3"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="FormatEnum">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Delimited"/>
      <xs:enumeration value="FixedWidth"/>
      <xs:enumeration value="RaggedRight"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="FlatFileColumnPropertyNameEnum">
    <xs:union memberTypes="DTS:BasePropertyNameEnum">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="ColumnType"/>
          <xs:enumeration value="ColumnDelimiter"/>
          <xs:enumeration value="ColumnWidth"/>
          <xs:enumeration value="MaximumWidth"/>
          <xs:enumeration value="DataType"/>
          <xs:enumeration value="DataPrecision"/>
          <xs:enumeration value="DataScale"/>
          <xs:enumeration value="TextQualified"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:union>
  </xs:simpleType>

  <xs:attributeGroup name="FlatFileColumnAttributeGroup">
    <xs:attribute name="ColumnType" type="DTS:ColumnTypeEnum"
                  default="FixedWidth" use="optional" form="qualified"/>
    <xs:attribute name="ColumnDelimiter" type="xs:string"
                  use="required" form="qualified"/>
    <xs:attribute name="ColumnWidth" type="xs:int" default="0"
                  use="optional" form="qualified"/>
    <xs:attribute name="MaximumWidth" type="xs:int" default="0"
                  use="optional" form="qualified"/>
    <xs:attribute name="DataType" type="DTS:DtsDataTypeEnum" default="0"
                  use="optional" form="qualified"/>
    <xs:attribute name="DataPrecision" type="xs:int" default="0"
                  use="optional" form="qualified"/>
    <xs:attribute name="DataScale" type="xs:int" default="0" use="optional"
                  form="qualified"/>
    <xs:attribute name="TextQualified" type="DTS:BooleanStringCap"
                  default="False" use="optional" form="qualified"/>
  </xs:attributeGroup>
  <!--1 ENUM follows-->
  <xs:simpleType name="ColumnTypeEnum">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Delimited"/>
      <xs:enumeration value="FixedWidth"/>
    </xs:restriction>
```

```
          </xs:simpleType>

     <xs:simpleType name="CacheColumnPropertyNameEnum">
       <xs:union memberTypes="DTS:BasePropertyNameEnum">
         <xs:simpleType>
           <xs:restriction base="xs:string">
             <xs:enumeration value="DataType"/>
             <xs:enumeration value="Length"/>
             <xs:enumeration value="Precision"/>
             <xs:enumeration value="Scale"/>
             <xs:enumeration value="CodePage"/>
             <xs:enumeration value="IndexPosition"/>
           </xs:restriction>
         </xs:simpleType>
       </xs:union>
     </xs:simpleType>

     <xs:attributeGroup name="CacheColumnAttributeGroup">
       <xs:attribute name="DataType" type="DTS:DtsDataTypeEnum" default="0"
                     use="optional" form="qualified"/>
       <xs:attribute name="Length" type="xs:int" default="0" use="optional"
                     form="qualified"/>
       <xs:attribute name="Precision" type="xs:int" default="0" use="optional"
                     form="qualified"/>
       <xs:attribute name="Scale" type="xs:int" default="0" use="optional"
                     form="qualified"/>
       <xs:attribute name="CodePage" type="xs:int" default="0" use="optional"
                     form="qualified"/>
       <xs:attribute name="IndexPosition" type="xs:int" default="0"
                     use="optional" form="qualified"/>
     </xs:attributeGroup>
     <!--0 ENUM-->

     <xs:simpleType name="FtpConnectionPropertyNameEnum">
       <xs:restriction base="xs:string">
         <xs:enumeration value="ServerPassword"/>
       </xs:restriction>
     </xs:simpleType>

     <xs:attributeGroup name="FtpConnectionAttributeGroup">
       <xs:attribute name="ServerName" type="xs:string" use="required"
                     form="qualified"/>
       <xs:attribute name="ServerPort" type="xs:int" default="21"
                     use="optional" form="qualified"/>
       <xs:attribute name="ServerUserName" type="xs:string" default=""
                     use="optional" form="qualified"/>
       <xs:attribute name="ChunkSize" type="xs:int" default="1" use="optional"
                     form="qualified"/>
       <xs:attribute name="TimeOut" type="xs:int" default="60" use="optional"
                     form="qualified"/>
       <xs:attribute name="Retries" type="xs:int" default="5" use="optional"
                     form="qualified"/>
       <xs:attribute name="UsePassiveMode" type="DTS:BooleanStringCap"
                     default="False" use="optional" form="qualified"/>
     </xs:attributeGroup>
     <!--0 ENUM-->

     <xs:simpleType name="HttpConnectionPropertyNameEnum">
       <xs:restriction base="xs:string">
         <xs:enumeration value="ProxyPassword"/>
         <xs:enumeration value="ServerPassword"/>
       </xs:restriction>
     </xs:simpleType>

     <xs:attributeGroup name="HttpConnectionAttributeGroup">
       <xs:attribute name="ServerURL" type="xs:string" use="required"
                     form="qualified"/>
       <xs:attribute name="UseServerCredentials" type="DTS:BooleanStringCap"
                     default="False" use="optional" form="qualified"/>
       <xs:attribute name="ServerUserName" type="xs:string" default=""
```

```
                     use="optional" form="qualified"/>
      <xs:attribute name="ServerDomain" type="xs:string" default=""
                     use="optional" form="qualified"/>
      <xs:attribute name="UseSecureConnection" type="DTS:BooleanStringCap"
                     default="False" use="optional" form="qualified"/>
      <xs:attribute name="CertificateIdentifier" type="xs:string" default=""
                     use="optional" form="qualified"/>
      <xs:attribute name="Certificate" type="xs:string" default=""
                     use="optional" form="qualified"/>
      <xs:attribute name="TimeOut" type="xs:int" default="60" use="optional"
                     form="qualified"/>
      <xs:attribute name="ChunkSize" type="xs:int" default="1" use="optional"
                     form="qualified"/>
      <xs:attribute name="UseProxy" type="DTS:BooleanStringCap"
                     default="False" use="optional" form="qualified"/>
      <xs:attribute name="ProxyURL" type="xs:string" default="" use="optional"
                     form="qualified"/>
      <xs:attribute name="BypassProxyOnLocal" type="DTS:BooleanStringCap"
                     default="False" use="optional" form="qualified"/>
      <xs:attribute name="UseProxyCredentials" type="DTS:BooleanStringCap"
                     default="False" use="optional" form="qualified"/>
      <xs:attribute name="ProxyUserName" type="xs:string" default=""
                     use="optional" form="qualified"/>
      <xs:attribute name="ProxyDomain" type="xs:string" default=""
                     use="optional" form="qualified"/>
      <xs:attribute name="BypassList" type="xs:string" default=""
                     use="optional" form="qualified"/>
  </xs:attributeGroup>

  <xs:simpleType name="EventHandlerPropertyNameEnum">
    <xs:union memberTypes="DTS:BaseExecutablePropertyNameEnum">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="EventID"/>
          <xs:enumeration value="EventName"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:union>
  </xs:simpleType>

  <xs:attributeGroup name="EventHandlerAttributeGroup">
    <xs:attribute name="EventID" type="xs:int" use="optional"
                   default="0" form="qualified"/>
    <xs:attribute name="EventName" type="DTS:EventNameEnum"
                   use="required" form="qualified"/>
  </xs:attributeGroup>

  <xs:simpleType name="EventNameEnum">
    <xs:restriction base="xs:string">
      <xs:enumeration value="OnError"/>
      <xs:enumeration value="OnExecStatusChanged"/>
      <xs:enumeration value="OnInformation"/>
      <xs:enumeration value="OnPostExecute"/>
      <xs:enumeration value="OnPreExecute"/>
      <xs:enumeration value="OnPreValidate"/>
      <xs:enumeration value="OnProgress"/>
      <xs:enumeration value="OnQueryCancel"/>
      <xs:enumeration value="OnTaskFailed"/>
      <xs:enumeration value="OnVariableValueChanged"/>
      <xs:enumeration value="OnWarning"/>
      <xs:enumeration value="OnCustomEvent"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="FileSystemTaskOperationTypeEnum">
    <xs:restriction base="xs:string">
      <xs:enumeration value="CopyFile" />
      <xs:enumeration value="CreateDirectory" />
      <xs:enumeration value="DeleteDirectory" />
      <xs:enumeration value="DeleteFile" />
```

```
          <xs:enumeration value="MoveDirectory" />
          <xs:enumeration value="MoveFile" />
          <xs:enumeration value="RenameFile" />
          <xs:enumeration value="SetAttributes" />
          <xs:enumeration value="DeleteDirectoryContent" />
          <xs:enumeration value="CopyDirectory" />
      </xs:restriction>
   </xs:simpleType>

   <xs:simpleType name="FileSystemTaskFileAttributesEnum">
     <xs:restriction base="xs:string">
       <xs:enumeration value="Archive" />
       <xs:enumeration value="Hidden" />
       <xs:enumeration value="Normal" />
       <xs:enumeration value="ReadOnly" />
       <xs:enumeration value="System" />
     </xs:restriction>
   </xs:simpleType>

   <xs:simpleType name="SourceTypeEnum">
     <xs:restriction base="xs:string">
       <xs:enumeration value="DirectInput"/>
       <xs:enumeration value="FileConnection"/>
       <xs:enumeration value="Variable"/>
     </xs:restriction>
   </xs:simpleType>

   <!--Other Misc Types-->

   <xs:simpleType name="uuid">
     <xs:restriction base="xs:string">
       <xs:pattern value="\{[0-9a-zA-Z]{8}-[0-9a-zA-Z]{4}-[0-9a-zA-Z]{4}-[0-9a-zA-Z]{4}-
           [0-9a-zA-Z]{12}\}" />
     </xs:restriction>
   </xs:simpleType>


</xs:schema>
```

### 5.1.2  DTSX2 Schema for Version 2014/01 (DTSX2 2014/01)

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:DTS="www.microsoft.com/SqlServer/Dts"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           elementFormDefault="qualified"
           attributeFormDefault="qualified"
           targetNamespace="www.microsoft.com/SqlServer/Dts"
           xmlns:WSTask="www.microsoft.com/sqlserver/dts/tasks/webservicetask"
           xmlns:MessageQueueTask="www.microsoft.com/sqlserver/dts/tasks/messagequeuetask"
           xmlns:SendMailTask="www.microsoft.com/sqlserver/dts/tasks/sendmailtask"
           xmlns:ActiveXScriptTask=
           "www.microsoft.com/sqlserver/dts/tasks/activexscripttask"
           xmlns:Exec80PackageTask=
           "www.microsoft.com/sqlserver/dts/tasks/exec80packagetask"
           xmlns:BulkInsertTask=
           "www.microsoft.com/sqlserver/dts/tasks/bulkinserttask"
           xmlns:SQLTask="www.microsoft.com/sqlserver/dts/tasks/sqltask"
           >

   <!--Imports-->

   <xs:import namespace=
        "www.microsoft.com/sqlserver/dts/tasks/webservicetask"
           schemaLocation="./ns_webservicetask.xsd"  />
   <xs:import namespace=
        "www.microsoft.com/sqlserver/dts/tasks/messagequeuetask"
           schemaLocation="./ns_messagequeuetask.xsd"    />
```

```
    <xs:import namespace=
        "www.microsoft.com/sqlserver/dts/tasks/sendmailtask"
            schemaLocation="./ns_sendmailtask.xsd" />
    <xs:import namespace=
        "www.microsoft.com/sqlserver/dts/tasks/activexscripttask"
            schemaLocation="./ns_activexscripttask.xsd"  />
    <xs:import namespace=
        "www.microsoft.com/sqlserver/dts/tasks/exec80packagetask"
            schemaLocation="./ns_exec80packagetask.xsd" />
    <xs:import namespace=
        "www.microsoft.com/sqlserver/dts/tasks/bulkinserttask"
                        schemaLocation="./ns_bulkinserttask.xsd" />
    <xs:import namespace=
        "www.microsoft.com/sqlserver/dts/tasks/sqltask"
            schemaLocation="./ns_sqltask.xsd" />


    <!--Root Element and Root Executables-->

    <xs:element name="Executable" type="DTS:ExecutableTypePackage"/>

      <xs:complexType name="ExecutableTypePackage">
        <xs:sequence>
          <xs:element name="Property" maxOccurs="unbounded">
            <xs:complexType>
              <xs:complexContent>
                <xs:extension base="DTS:PropertyElementBaseType">
                  <xs:attribute name="Name"
                        type="DTS:ExecutableTypePackagePropertyNameEnum"
                        use="required"/>
                </xs:extension>
              </xs:complexContent>
            </xs:complexType>
          </xs:element>
          <xs:element name="ConnectionManagers" type="DTS:ConnectionManagersType"
                    minOccurs="0"/>
          <xs:element name="Configurations" type="DTS:ConfigurationsType"
                    minOccurs="0"/>
          <xs:element name="LogProviders" type="DTS:LogProvidersType"
                    minOccurs="0"/>
          <xs:element name="Variables" type="DTS:VariablesType"
                    minOccurs="0"/>
          <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType" minOccurs="0"/>
          <xs:element name="PropertyExpression"
                    type="DTS:PropertyExpressionElementType"
                    minOccurs="0" maxOccurs="unbounded"/>
          <xs:element name="Executables" type="DTS:ExecutablesType"
                    minOccurs="0"/>
          <xs:element name="PrecedenceConstraints" minOccurs="0"
                    type="DTS:PrecedenceConstraintsType"/>
          <xs:element name="EventHandlers" type="DTS:EventHandlersType"
                    minOccurs="0"/>
          <xs:element name="PackageParameters" type="DTS:PackageParametersType"
                    minOccurs="0" />
          <xs:element name="PackageVariables" type="DTS:PackageVariablesType"
                    minOccurs="0" maxOccurs="unbounded"/>
          <xs:element name="DesignTimeProperties" type="xs:string"
                    minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="ExecutableType" use="required"
                fixed="SSIS.Package.3" />
        <xs:attribute name="refId" type="xs:string" use="required"
                    form="qualified"/>
        <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
        <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
        <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
        <xs:attributeGroup ref="DTS:ExecutableTypePackageAttributeGroup"/>
      </xs:complexType>

    <xs:complexType name="ExecutablesType">
```

```
      <xs:sequence>
        <xs:element name="Executable" type="DTS:AnyNonPackageExecutableType"
                    minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>

    <xs:complexType name="AnyNonPackageExecutableType">
      <xs:sequence>
        <xs:element name="ForEachEnumerator"
                    type="DTS:ForEachEnumeratorType" minOccurs="0"/>
        <xs:element name="Variables" type="DTS:VariablesType"
                    minOccurs="0"/>
        <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType"
                    minOccurs="0"/>
        <xs:element name="PropertyExpression"
                    type="DTS:PropertyExpressionElementType"
                    minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="Executables" type="DTS:ExecutablesType"
                    minOccurs="0"/>
        <xs:element name="PrecedenceConstraints"
                    type="DTS:PrecedenceConstraintsType"
                    minOccurs="0"/>
        <xs:element name="ForEachVariableMappings"
                    type="DTS:ForEachVariableMappingsType" minOccurs="0"/>
        <xs:element name="EventHandlers" type="DTS:EventHandlersType"
                    minOccurs="0"/>
        <xs:element name="ObjectData" type="DTS:ExecutableObjectDataType"
                    minOccurs="0"/>
      </xs:sequence>
      <xs:attribute name="ExecutableType" use="required" type="xs:string"/>
      <xs:attribute name="ThreadHint" use="optional" type="xs:int"/>
      <xs:attributeGroup ref="DTS:AllExecutableAttributeGroup"/>
      <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
      <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
      <xs:attributeGroup ref="DTS:AnyNonPackageExecutableAttributeGroup"/>
      <xs:attribute name="refId" type="xs:string" use="required"
                    form="qualified"/>
    </xs:complexType>

    <!--Package Exe contained types-->

    <xs:complexType name="PackageParametersType">
      <xs:sequence>
        <xs:element name="PackageParameter" type="DTS:PackageParameterType" minOccurs="0"
                    maxOccurs="unbounded" />
      </xs:sequence>
    </xs:complexType>

    <xs:complexType name="PackageParameterType">
      <xs:sequence>
        <xs:element maxOccurs="unbounded" name="Property"
                    type="DTS:PackageParameterPropertyType" />
      </xs:sequence>
      <xs:attribute name="CreationName" use="required" type="xs:string" />
      <xs:attribute name="DataType" type="xs:unsignedByte" use="required" />
      <xs:attribute name="Description" type="xs:string" use="optional" />
      <xs:attribute name="DTSID" type="DTS:uuid" use="required" />
      <xs:attribute name="ObjectName" type="xs:string" use="required" />
      <xs:attribute name="Required" type="xs:string" use="optional" />
      <xs:attribute name="Sensitive" type="xs:string" use="optional" />
    </xs:complexType>

    <xs:complexType mixed="true" name="PackageParameterPropertyType">
      <xs:sequence minOccurs="0">
        <xs:element name="Property"
                    type="DTS:PackageParameterPropertyEncryptedValuePropertyType" />
      </xs:sequence>
      <xs:attribute name="DataType" type="xs:unsignedByte" use="optional" />
      <xs:attribute name="Name" type="xs:string" use="required" />
    </xs:complexType>
```

```
<xs:complexType name="PackageParameterPropertyEncryptedValuePropertyType">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="DataType" type="xs:unsignedByte" use="required" />
      <xs:attribute name="Name" type="xs:string" use="required" />
      <xs:attribute name="Encrypted" type="xs:unsignedByte" use="required" />
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:complexType name="PackageVariablesType">
  <xs:sequence>
    <xs:element name="PackageVariable" type="DTS:PackageVariableType"
                minOccurs="0" maxOccurs="unbounded" form="qualified"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="PackageVariableType">
  <xs:sequence>
    <xs:element name="Property"
                maxOccurs="unbounded">
      <xs:complexType>
        <xs:complexContent>
          <xs:extension base="DTS:PropertyElementBaseType">
            <xs:attribute name="Name"
                          type="DTS:PackageVariablePropertyNameEnum"
                          use="required"/>
          </xs:extension>
        </xs:complexContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="PropertyExpression"
                type="DTS:PropertyExpressionElementType"
                minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="ConfigurationsType">
  <xs:sequence>
    <xs:element name="Configuration" type="DTS:ConfigurationType"
                minOccurs="0" maxOccurs="unbounded" form="qualified"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="ConfigurationType">
  <xs:sequence>
    <xs:element name="PropertyExpression"
                type="DTS:PropertyExpressionElementType"
                minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
  <xs:attributeGroup ref="DTS:ConfigurationAttributeGroup"/>
</xs:complexType>

<xs:complexType name="LogProvidersType">
  <xs:sequence>
    <xs:element name="LogProvider" type="DTS:LogProviderType"
                minOccurs="0" maxOccurs="unbounded"  form="qualified"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="LogProviderType" >
  <xs:sequence>
    <xs:element name="PropertyExpression"
                type="DTS:PropertyExpressionElementType"
                minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="ObjectData" type="DTS:LogProviderObjectDataType"/>
  </xs:sequence>
  <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
  <xs:attributeGroup ref="DTS:LogProviderAttributeGroup"/>
```

```xml
    </xs:complexType>

    <xs:complexType name="ConnectionManagersType">
      <xs:sequence>
        <xs:element name="ConnectionManager" type="DTS:ConnectionManagerType"
                    minOccurs="0" maxOccurs="unbounded" form="qualified"/>
      </xs:sequence>
    </xs:complexType>

    <xs:complexType name="ConnectionManagerType">
      <xs:sequence>
        <xs:element name="PropertyExpression"
                    type="DTS:PropertyExpressionElementType"
                    minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="ObjectData"
                    type="DTS:ConnectionManagerObjectDataType"/>
      </xs:sequence>
      <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
      <xs:attributeGroup ref="DTS:ConnectionManagerAttributeGroup"/>
      <xs:attribute name="refId" type="xs:string" use="required"
                    form="qualified"/>
    </xs:complexType>

    <!--AnyNonPackageExecutable Contained Complex Types-->

    <xs:complexType name="PrecedenceConstraintsType">
      <xs:sequence>
        <xs:element name="PrecedenceConstraint"
                    type="DTS:PrecedenceConstraintType"
                    minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>

    <xs:complexType name="PrecedenceConstraintType">
      <xs:sequence>
        <xs:element name="PropertyExpression"
                    type="DTS:PropertyExpressionElementType"
                    minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
      <xs:attributeGroup ref="DTS:PrecedenceConstraintAttributeGroup"/>
      <xs:attribute name="refId" type="xs:string" use="required"
                    form="qualified"/>
    </xs:complexType>

    <xs:complexType name="LoggingOptionsType">
      <xs:sequence>
        <xs:element name="Property"
                    type="DTS:LoggingOptionsPropertyElementType"
                    minOccurs="0" maxOccurs="unbounded">
        </xs:element>
        <xs:element name="PropertyExpression"
                    type="DTS:PropertyExpressionElementType"
                    minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="SelectedLogProviders" minOccurs="0"
                    type="DTS:SelectedLogProvidersType"/>
      </xs:sequence>
      <xs:attributeGroup ref="DTS:LoggingOptionsAttributeGroup"/>
    </xs:complexType>

    <xs:complexType name="LoggingOptionsPropertyElementType" mixed="true">
      <xs:sequence>
        <xs:element name="Property" minOccurs="0" maxOccurs="unbounded">
          <xs:complexType>
            <xs:simpleContent>
              <xs:extension base="xs:anySimpleType">
                <xs:attribute name="Name"
                              type="DTS:ColumnEventFilterPropertyNameEnum"
                              use="required"/>
                <xs:attribute name="DataType" type="DTS:DtsDataTypeEnum"/>
```

```
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="Name"
      type="DTS:LoggingOptionsPropertyNameEnum"
      use="required"/>
  <xs:attribute name="DataType" type="DTS:DtsDataTypeEnum"/>
  <xs:attribute name="EventName" type="xs:string"/>
</xs:complexType>

<xs:complexType name="SelectedLogProvidersType">
  <xs:sequence>
    <xs:element name="SelectedLogProvider"
              type="DTS:SelectedLogProviderType"
              minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="SelectedLogProviderType">
  <xs:sequence/>
  <xs:attribute name="InstanceID" type="DTS:uuid"/>
</xs:complexType>

<xs:complexType name="VariablesType">
  <xs:sequence>
    <xs:element name="Variable" type="DTS:VariableType"
              minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="VariableType">
  <xs:sequence>
    <xs:element name="PropertyExpression"
              type="DTS:PropertyExpressionElementType"
              minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="VariableValue">
      <xs:complexType mixed="true">
        <xs:sequence>
          <xs:any minOccurs="0" maxOccurs="unbounded"
                processContents="skip"/>
        </xs:sequence>
        <xs:attribute name="DataType" type="DTS:DtsDataTypeEnum"/>
        <xs:attribute name="DataSubType">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="ManagedSerializable"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
  <xs:attributeGroup ref="DTS:VariableAttributeGroup"/>
</xs:complexType>

<xs:complexType name="EventHandlersType">
  <xs:sequence>
    <xs:element name="EventHandler" type="DTS:EventHandlerType"
              minOccurs="0" maxOccurs="unbounded" form="qualified"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="EventHandlerType">
  <xs:sequence>
    <xs:element name="PropertyExpression"
              type="DTS:PropertyExpressionElementType"
              minOccurs="0" maxOccurs="unbounded"/>
```

```
          <xs:element name="Variables" type="DTS:VariablesType" minOccurs="0"/>
          <xs:element name="LoggingOptions" type="DTS:LoggingOptionsType"/>
          <xs:element name="Executables" type="DTS:ExecutablesType" minOccurs="0"/>
          <xs:element name="PrecedenceConstraints"
                      type="DTS:PrecedenceConstraintsType" minOccurs="0"/>
      </xs:sequence>
      <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
      <xs:attributeGroup ref="DTS:BaseExecutablePropertyAttributeGroup"/>
      <xs:attributeGroup ref="DTS:EventHandlerAttributeGroup"/>
      <xs:attribute name="refId" type="xs:string" use="required"
                    form="qualified"/>
  </xs:complexType>

  <xs:complexType name="ForEachEnumeratorType">
    <xs:sequence>
      <xs:element name="PropertyExpression"
                  type="DTS:PropertyExpressionElementType"
                  minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="ObjectData"
                  type="DTS:ForEachEnumeratorObjectDataType"/>
    </xs:sequence>
    <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
  </xs:complexType>

  <xs:complexType name="ForEachVariableMappingsType">
    <xs:sequence>
      <xs:element name="ForEachVariableMapping"
                  type="DTS:ForEachVariableMappingType"
                  minOccurs="0" maxOccurs="unbounded" form="qualified"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="ForEachVariableMappingType" >
    <xs:sequence>
      <xs:element name="PropertyExpression"
                  type="DTS:PropertyExpressionElementType"
                  minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
    <xs:attributeGroup ref="DTS:ForEachVariableMappingAttributeGroup"/>
  </xs:complexType>

  <!--ObjectData Executable Types-->

  <xs:complexType name="ExecutableObjectDataType">
    <xs:choice minOccurs="0">
      <xs:element name="pipeline" type="DTS:pipelineObjectDataType"
                  form="unqualified"/>
      <xs:element name="CDCControlTask" form="unqualified"
                  type="DTS:CDCControlTaskObjectDataType"/>
      <xs:element name="ASDdlData" type="DTS:ASDdlDataObjectDataType"
                  form="unqualified"/>
      <xs:element name="ASProcessingData" form="unqualified"
                  type="DTS:ASProcessingDataObjectDataType"/>
      <xs:element ref="BulkInsertTask:BulkInsertTaskData"/>
      <xs:element name="DMQueryTaskData"  form="unqualified"
                  type="DTS:DMQueryTaskDataObjectDataType"/>
      <xs:element name="DataProfilingTaskData"  form="unqualified"
                  type="DTS:DataProfilingTaskDataObjectDataType"/>
      <xs:element name="ExecutePackageTask"  form="unqualified"
                  type="DTS:ExecutePackageTaskObjectDataType"/>
      <xs:element name="ExecuteProcessData" form="unqualified"
                  type="DTS:ExecuteProcessDataObjectDataType"/>
      <xs:element name="ExpressionTask" form="unqualified"
                  type="DTS:ExpressionTaskObjectDataType"/>
      <xs:element ref="SQLTask:SqlTaskData"/>
      <xs:element name="FileSystemData" form="unqualified"
                  type="DTS:FileSystemDataObjectDataType"/>
      <xs:element name="FtpData" type="DTS:FtpDataObjectDataType"
                  form="unqualified"/>
```

```
            <xs:element ref="MessageQueueTask:MessageQueueTaskData"/>
            <xs:element name="ScriptProject" type="DTS:ScriptProjectObjectDataType"
                        form="unqualified"/>
            <xs:element ref="SendMailTask:SendMailTaskData"/>
            <xs:element name="TransferDatabasesTaskData" form="unqualified"
                        type="DTS:TransferDatabasesTaskDataObjectDataType"/>
            <xs:element name="TransferErrorMessagesTaskData" form="unqualified"
                        type="DTS:TransferErrorMessagesTaskDataObjectDataType"/>
            <xs:element name="TransferJobsTaskData" form="unqualified"
                        type="DTS:TransferJobsTaskDataObjectDataType"/>
            <xs:element name="TransferLoginsTaskData" form="unqualified"
                        type="DTS:TransferLoginsTaskDataObjectDataType"/>
            <xs:element name="TransferStoredProceduresTaskData" form="unqualified"
                        type="DTS:TransferStoredProceduresTaskDataObjectDataType"/>
            <xs:element name="TransferSqlServerObjectsTaskData" form="unqualified"
                        type="DTS:TransferSqlServerObjectsTaskDataObjectDataType"/>
            <xs:element ref="WSTask:WebServiceTaskData"/>
            <xs:element name="WMIDRTaskData" type="DTS:WMIDRTaskDataObjectDataType"
                        form="unqualified"/>
            <xs:element name="WMIEWTaskData" type="DTS:WMIEWTaskDataObjectDataType"
                        form="unqualified"/>
            <xs:element name="XMLTaskData" type="DTS:XMLTaskDataObjectDataType"
                        form="unqualified"/>
            <xs:element name="InnerObject" type="DTS:InnerObjectObjectDataType"
                        form="unqualified"/>
        </xs:choice>
    </xs:complexType>

    <xs:complexType name="ASDdlDataObjectDataType">
        <xs:sequence/>
        <xs:attribute name="ConnectionName" type="xs:string"
                        form="unqualified"/>
        <xs:attribute name="SourceType" form="unqualified"
                        type="DTS:SourceTypeEnum"/>
        <xs:attribute name="Source" type="xs:string"
                        form="unqualified"/>
    </xs:complexType>

    <xs:complexType name="ASProcessingDataObjectDataType">
        <xs:sequence/>
        <xs:attribute name="ConnectionName" type="xs:string"
                        form="unqualified"/>
        <xs:attribute name="ProcessingCommands" type="xs:string"
                        form="unqualified"/>
    </xs:complexType>

    <xs:complexType name="DMQueryTaskDataObjectDataType">
        <xs:sequence>
            <xs:element name="QueryString" form="unqualified"
                        type="DTS:DMQueryTaskQueryStringContainerType"/>
            <xs:element name="QueryBuilderQueryString" form="unqualified"
                        type="DTS:DMQueryTaskQueryStringContainerType"/>
            <xs:element name="QueryBuilderSpecification" form="unqualified"
                        type="DTS:DMQueryBuilderSpecificationType"/>
        </xs:sequence>
        <xs:attribute name="InputConnection" type="xs:string"
                        form="unqualified"/>
        <xs:attribute name="OutputConnection" type="xs:string"
                        form="unqualified"/>
        <xs:attribute name="ModelStructureName" type="xs:string"
                        form="unqualified"/>
        <xs:attribute name="ModelName" type="xs:string" form="unqualified"/>
        <xs:attribute name="OverwriteOutputTable" form="unqualified">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:enumeration value="True"/>
                    <xs:enumeration value="False"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
```

```
        <xs:attribute name="OutputTableName" type="xs:string"
                    form="unqualified"/>
  </xs:complexType>

  <xs:complexType name="DMQueryTaskQueryStringContainerType">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="Sensitive" type="xs:boolean"
                    form="unqualified"/>
        <xs:attribute name="Encrypted" type="xs:boolean"
                    form="unqualified"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>

  <xs:complexType name="DMQueryBuilderSpecificationType">
    <xs:sequence>
      <xs:any minOccurs="0" maxOccurs="unbounded"
            namespace="##any" processContents="lax"/>
    </xs:sequence>
    <xs:attribute name="Sensitive" type="xs:boolean"
                    form="unqualified"/>
    <xs:attribute name="Encrypted" type="xs:boolean"
                form="unqualified"/>
  </xs:complexType>

  <xs:complexType name="DataProfilingTaskDataObjectDataType">
    <xs:sequence>
      <xs:element name="ProfileInput" type="xs:string"
                form="unqualified"/>
    </xs:sequence>
    <xs:attribute name="DestinationType" form="unqualified"
                type="DTS:DestinationTypeEnum"
                use="optional" default="FileConnection"/>
    <xs:attribute name="Destination" type="xs:string"
                form="unqualified" use="optional" default=""/>
    <xs:attribute name="OverwriteDestination" form="unqualified"
                use="optional" default="False">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="True"/>
          <xs:enumeration value="False"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:complexType>

  <xs:simpleType name="DestinationTypeEnum">
    <xs:restriction base="xs:string">
      <xs:enumeration value="FileConnection"/>
      <xs:enumeration value="Variable"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="ExecutePackageTaskObjectDataType">
    <xs:sequence>
      <xs:element name="ExecuteOutOfProcess" form="unqualified">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="True"/>
            <xs:enumeration value="False"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="PackageName" type="xs:string"
                form="unqualified"/>
      <xs:element name="UseProjectReference" type="DTS:BooleanStringCap"
                minOccurs="0" form="unqualified"/>
      <xs:element name="PackagePassword" type="DTS:PasswordElementType"
                form="unqualified"/>
```

```
        <xs:element name="PackageID" type="DTS:uuid" form="unqualified"/>
        <xs:element name="VersionID" type="DTS:uuid" form="unqualified"/>
        <xs:element name="Connection" type="xs:string" form="unqualified"/>
      </xs:sequence>
    </xs:complexType>

    <xs:complexType name="ExecuteProcessDataObjectDataType">
      <xs:sequence>
      </xs:sequence>
      <xs:attribute name="Executable" type="xs:string" form="unqualified"
                    use="optional" default=""/>
      <xs:attribute name="RequireFullFileName" form="unqualified"
                    use="optional" type="DTS:BooleanStringCap"
                    default="True"/>
      <xs:attribute name="Arguments" type="xs:string" form="unqualified"
                    use="optional" default=""/>
      <xs:attribute name="WorkingDirectory" type="xs:string"
                    use="optional" default="" form="unqualified"/>
      <xs:attribute name="TimeOut" type="xs:int" form="unqualified"
                    use="optional" default="0"/>
      <xs:attribute name="FailTaskIfReturnCodeIsNotSuccessValue"
                    type="DTS:BooleanStringCap" use="optional" default="True"
                    form="unqualified">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="True"/>
            <xs:enumeration value="False"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="SuccessValue" type="xs:int" form="unqualified"
                    use="optional" default="0"/>
      <xs:attribute name="TerminateAfterTimeout" form="unqualified"
                    use="optional" default="True">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="True"/>
            <xs:enumeration value="False"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="StandardInputVariable" type="xs:string"
                    use="optional" default="" form="unqualified"/>
      <xs:attribute name="StandardOutputVariable" type="xs:string"
                    use="optional" default="" form="unqualified"/>
      <xs:attribute name="StandardErrorVariable" type="xs:string"
                    use="optional" default="" form="unqualified"/>
      <xs:attribute name="WindowStyle" form="unqualified"
                    type="DTS:WindowStyleEnum"
                    use="optional" default="Normal"/>
    </xs:complexType>

    <xs:simpleType name="WindowStyleEnum">
      <xs:restriction base="xs:string">
        <xs:enumeration value="Hidden"/>
        <xs:enumeration value="Maximized"/>
        <xs:enumeration value="Minimized"/>
        <xs:enumeration value="Normal"/>
      </xs:restriction>
    </xs:simpleType>

    <xs:complexType name="FileSystemDataObjectDataType">
      <xs:sequence/>
      <xs:attribute name="TaskOperationType" form="unqualified"
                    type="DTS:FileSystemTaskOperationTypeEnum"
                    use="optional" default="CopyFile"/>
      <xs:attribute name="OperationName" form="unqualified"
                    type="xs:string" use="optional" default=""/>
      <xs:attribute name="TaskOverwriteDestFile" form="unqualified"
                    use="optional" default="False">
```

```
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="False"/>
            <xs:enumeration value="True"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="TaskSourcePath" form="unqualified"
                    type="xs:string" use="optional" default=""/>
      <xs:attribute name="TaskIsSourceVariable" form="unqualified"
                    use="optional" default="False">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="False"/>
            <xs:enumeration value="True"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="TaskDestinationPath" form="unqualified"
                    type="xs:string" use="optional" default=""/>
      <xs:attribute name="TaskIsDestinationVariable" form="unqualified">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="False"/>
            <xs:enumeration value="True"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="TaskFileAttributes" form="unqualified"
                    type="DTS:FileSystemTaskFileAttributesEnum"
                    use="optional" default="Normal"/>
  </xs:complexType>

  <xs:complexType name="FtpDataObjectDataType">
    <xs:sequence>
    </xs:sequence>
    <xs:attribute name="ConnectionName" type="xs:string"
                  use="optional" default="" form="unqualified"/>
    <xs:attribute name="StopOnFailure" form="unqualified">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="False"/>
          <xs:enumeration value="True"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="TaskOperationType"
                  type="DTS:FtpDataTaskOperationTypeEnum"
                  use="optional" default="Send" form="unqualified"/>
    <xs:attribute name="TaskOperationName" type="xs:string"
                  use="optional" default="" form="unqualified"/>
    <xs:attribute name="TaskIsLocalPathVar" form="unqualified"
                  use="optional" default="False">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="False"/>
          <xs:enumeration value="True"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="TaskLocalPath" type="xs:string"
                  use="optional" default="" form="unqualified"/>
    <xs:attribute name="TaskIsRemotePathVar" form="unqualified"
                  use="optional" default="False">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="False"/>
          <xs:enumeration value="True"/>
        </xs:restriction>
      </xs:simpleType>
```

```
    </xs:attribute>
    <xs:attribute name="TaskRemotePath" type="xs:string"
                  use="optional" default="" form="unqualified"/>
    <xs:attribute name="TaskOverwriteDest" form="unqualified"
                  use="optional" default="False">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="False"/>
          <xs:enumeration value="True"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="TaskTransferAscii" form="unqualified"
                  use="optional" default="False">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="False"/>
          <xs:enumeration value="True"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
</xs:complexType>

<xs:simpleType name="FtpDataTaskOperationTypeEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="DeleteLocal"/>
    <xs:enumeration value="DeleteRemote"/>
    <xs:enumeration value="MakeDirLocal"/>
    <xs:enumeration value="MakeDirRemote"/>
    <xs:enumeration value="Receive"/>
    <xs:enumeration value="RemoveDirLocal"/>
    <xs:enumeration value="RemoveDirRemote"/>
    <xs:enumeration value="Send"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="ScriptProjectObjectDataType">
  <xs:sequence>
    <xs:element name="ProjectItem"
                type="DTS:ScriptProjectProjectItemType"
                minOccurs="0" maxOccurs="unbounded"
                form="unqualified"/>
    <xs:element name="BinaryItem"
                type="DTS:ScriptProjectBinaryItemType"
                minOccurs="0" form="unqualified"/>
    <xs:element name="Breakpoint"
                type="DTS:ScriptProjectBreakpointType"
                minOccurs="0" maxOccurs="unbounded"
                form="unqualified"/>
  </xs:sequence>
  <xs:attribute name="Name" type="xs:string" form="unqualified"/>
  <xs:attribute name="VSTAMajorVersion" type="xs:int"
                form="unqualified"/>
  <xs:attribute name="VSTAMinorVersion" type="xs:int"
                form="unqualified"/>
  <xs:attribute name="Language" type="DTS:ScriptProjectLanguageEnum"
                form="unqualified"/>
  <xs:attribute name="EntryPoint" type="xs:string"
                form="unqualified"/>
  <xs:attribute name="ReadOnlyVariables" type="xs:string"
                form="unqualified"/>
  <xs:attribute name="ReadWriteVariables" type="xs:string"
                form="unqualified"/>
</xs:complexType>

<xs:complexType name="ScriptProjectProjectItemType">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="Name" type="xs:string"
                    form="unqualified"/>
```

```
            <xs:attribute name="Encoding" type="xs:string"
                        form="unqualified"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>

<xs:complexType name="ScriptProjectBinaryItemType">
    <xs:simpleContent>
        <xs:extension base="xs:base64Binary">
            <xs:attribute name="Name" type="xs:string"
                        form="unqualified"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>

<xs:complexType name="ScriptProjectBreakpointType">
    <xs:sequence/>
    <xs:attribute name="Name" type="xs:string" form="unqualified"/>
    <xs:attribute name="Enabled" type="xs:string"
                form="unqualified"/>
    <xs:attribute name="Function" type="xs:string"
                form="unqualified"/>
    <xs:attribute name="File" type="xs:string" form="unqualified"/>
    <xs:attribute name="Line" type="xs:int" form="unqualified"/>
    <xs:attribute name="Column" type="xs:int" form="unqualified"/>
    <xs:attribute name="Condition" type="xs:string"
                form="unqualified"/>
    <xs:attribute name="ConditionType"
                type="DTS:ScriptProjectConditionTypeEnum"
                form="unqualified"/>
    <xs:attribute name="Language" type="xs:string"
                form="unqualified"/>
    <xs:attribute name="HitCount" type="xs:int" form="unqualified"/>
    <xs:attribute name="HitCountType" form="unqualified"
                type="DTS:ScriptProjectHitCountTypeEnum"/>
</xs:complexType>

<xs:simpleType name="ScriptProjectLanguageEnum">
    <xs:restriction base="xs:string">
        <xs:enumeration value="CSharp"/>
        <xs:enumeration value="VisualBasic"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="ScriptProjectHitCountTypeEnum">
    <xs:restriction base="xs:int">
        <xs:minInclusive value="1"/>
        <xs:maxInclusive value="4"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="ScriptProjectConditionTypeEnum">
    <xs:restriction base="xs:int">
        <xs:minInclusive value="1"/>
        <xs:maxInclusive value="2"/>
    </xs:restriction>
</xs:simpleType>

<xs:complexType name="TransferDatabasesTaskDataObjectDataType">
    <xs:sequence/>
    <xs:attribute name="SrcConn" type="xs:string" use="optional" default=""
                form="unqualified"/>
    <xs:attribute name="DestConn" type="xs:string" use="optional"
                default="" form="unqualified"/>
    <xs:attribute name="Action" type="DTS:TransferDatabaseTaskActionEnum"
                use="optional" default="0" form="unqualified"/>
    <xs:attribute name="Method" type="DTS:TransferDatabaseTaskMethodEnum"
                use="optional" default="0" form="unqualified"/>
    <xs:attribute name="SrcDBName" type="xs:string" use="optional"
                default="" form="unqualified"/>
```

```
            <xs:attribute name="ReattachSrcDB" type="xs:boolean"
                       use="optional" default="false" form="unqualified"/>
            <xs:attribute name="SrcDBFiles" type="xs:string"
                       use="optional" default="" form="unqualified"/>
            <xs:attribute name="DestDBName" type="xs:string"
                       use="optional" default="" form="unqualified"/>
            <xs:attribute name="DestDBFiles" type="xs:string"
                       use="optional" default="" form="unqualified"/>
            <xs:attribute name="DestOverwrite" type="xs:boolean"
                       use="optional" default="false" form="unqualified"/>
      </xs:complexType>

      <xs:simpleType name="TransferDatabaseTaskActionEnum">
        <xs:restriction base="xs:int">
          <xs:minInclusive value="0"/>
          <xs:maxInclusive value="1"/>
        </xs:restriction>
      </xs:simpleType>

      <xs:simpleType name="TransferDatabaseTaskMethodEnum">
        <xs:restriction base="xs:int">
          <xs:minInclusive value="0"/>
          <xs:maxInclusive value="1"/>
        </xs:restriction>
      </xs:simpleType>

      <xs:complexType name="TransferErrorMessagesTaskDataObjectDataType">
        <xs:sequence/>
        <xs:attribute name="TransferAllErrorMessages" type="xs:boolean"
                       use="optional" default="false" form="unqualified"/>
        <xs:attribute name="ErrorMessagesList" type="xs:string"
                       use="optional" default="" form="unqualified"/>
        <xs:attribute name="ErrorMessageLanguagesList" type="xs:string"
                       use="optional" default="" form="unqualified"/>
        <xs:attribute name="IfObjectExists"
                       type="DTS:IfObjectExistsEnum"
                       use="optional" default="FailTask" form="unqualified"/>
        <xs:attribute name="SourceConnection" type="xs:string"
                       use="optional" default="" form="unqualified"/>
        <xs:attribute name="DestinationConnection" type="xs:string"
                       use="optional" default="" form="unqualified"/>
      </xs:complexType>

      <xs:simpleType name="IfObjectExistsEnum">
        <xs:restriction base="xs:string">
          <xs:enumeration value="FailTask"/>
          <xs:enumeration value="Overwrite"/>
          <xs:enumeration value="Skip"/>
        </xs:restriction>
      </xs:simpleType>

      <xs:complexType name="TransferJobsTaskDataObjectDataType">
        <xs:sequence/>
        <xs:attribute name="TransferAllJobs" type="xs:boolean"
                       use="optional" default="false" form="unqualified"/>
        <xs:attribute name="JobsList" type="xs:string"
                       use="optional" default="" form="unqualified"/>
        <xs:attribute name="IfObjectExists" type="DTS:IfObjectExistsEnum"
                       use="optional" default="FailTask" form="unqualified"/>
        <xs:attribute name="EnableJobsAtDestination" type="xs:boolean"
                       use="optional" default="false" form="unqualified"/>
        <xs:attribute name="SourceConnection" type="xs:string"
                       use="optional" default="" form="unqualified"/>
        <xs:attribute name="DestinationConnection" type="xs:string"
                       use="optional" default="" form="unqualified"/>
      </xs:complexType>

      <xs:complexType name="TransferLoginsTaskDataObjectDataType">
        <xs:sequence/>
        <xs:attribute name="LoginsToTransfer"
```

```xml
                    type="DTS:TransferLoginsTaskLoginsToTransferEnum"
                    use="optional" default="SelectedLogins"
                    form="unqualified"/>
    <xs:attribute name="CopySids" type="xs:boolean" form="unqualified"
                    use="optional" default="false"/>
    <xs:attribute name="LoginList" type="xs:string" form="unqualified"
                    use="optional" default=""/>
    <xs:attribute name="IfObjectExists" type="DTS:IfObjectExistsEnum"
                    use="optional" default="FailTask" form="unqualified"/>
    <xs:attribute name="SourceConnection" type="xs:string"
                    use="optional" default="" form="unqualified"/>
    <xs:attribute name="DestinationConnection" type="xs:string"
                    use="optional" default="" form="unqualified"/>
    <xs:attribute name="DatabasesList" type="xs:string"
                    use="optional" default="" form="unqualified"/>
  </xs:complexType>

  <xs:simpleType name="TransferLoginsTaskLoginsToTransferEnum">
    <xs:restriction base="xs:string">
      <xs:enumeration value="SelectedLogins"/>
      <xs:enumeration value="AllLoginsFromSelectedDatabases"/>
      <xs:enumeration value="AllLogins"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="TransferStoredProceduresTaskDataObjectDataType">
    <xs:sequence/>
    <xs:attribute name="TransferAllStoredProcedures"
                    type="xs:boolean" use="optional" default="false"
                    form="unqualified"/>
    <xs:attribute name="StoredProceduresList" type="xs:string"
                    use="optional" default="" form="unqualified"/>
    <xs:attribute name="IfObjectExists" type="DTS:IfObjectExistsEnum"
                    use="optional" default="FailTask" form="unqualified"/>
    <xs:attribute name="SourceConnection" type="xs:string"
                    use="optional" default="" form="unqualified"/>
    <xs:attribute name="m_DestinationConnectionID"
                    use="optional" default="" type="xs:string"
                    form="unqualified"/>
  </xs:complexType>

  <xs:complexType name="TransferSqlServerObjectsTaskDataObjectDataType">
    <xs:sequence/>
    <xs:attribute name="SourceConnection" type="xs:string"
                    use="optional" default="" form="unqualified"/>
    <xs:attribute name="DestinationConnection" type="xs:string"
                    use="optional" default="" form="unqualified"/>
    <xs:attribute name="SourceDatabase" type="xs:string"
                    use="optional" default="" form="unqualified"/>
    <xs:attribute name="DestinationDatabase" type="xs:string"
                    use="optional" default="" form="unqualified"/>
    <xs:attribute name="CopyAllObjects" type="xs:boolean"
                    use="optional" default="false" form="unqualified"/>
    <xs:attribute name="CopyAllRules" type="xs:boolean"
                    use="optional" default="false" form="unqualified"/>
    <xs:attribute name="CopyAllTables" type="xs:boolean"
                    use="optional" default="false" form="unqualified"/>
    <xs:attribute name="CopyAllViews" type="xs:boolean"
                    use="optional" default="false" form="unqualified"/>
    <xs:attribute name="CopyAllStoredProcedures"
                    type="xs:boolean" use="optional" default="false"
                    form="unqualified"/>
    <xs:attribute name="CopyAllUserDefinedFunctions"
                    type="xs:boolean" use="optional" default="false"
                    form="unqualified"/>
    <xs:attribute name="CopyAllDefaults" type="xs:boolean"
                    use="optional" default="false" form="unqualified"/>
    <xs:attribute name="CopyAllUserDefinedDataTypes"
                    use="optional" default="false" type="xs:boolean"
                    form="unqualified"/>
```

```xml
<xs:attribute name="CopyAllPartitionFunctions" type="xs:boolean"
              use="optional" default="false" form="unqualified"/>
<xs:attribute name="CopyAllPartitionSchemes" type="xs:boolean"
              use="optional" default="false" form="unqualified"/>
<xs:attribute name="CopyAllSchemas" type="xs:boolean"
              use="optional" default="false" form="unqualified"/>
<xs:attribute name="CopyAllSqlAssemblies" type="xs:boolean"
              use="optional" default="false" form="unqualified"/>
<xs:attribute name="CopyAllUserDefinedAggregates"
              use="optional" default="false" type="xs:boolean"
              form="unqualified"/>
<xs:attribute name="CopyAllUserDefinedTypes" type="xs:boolean"
              use="optional" default="false" form="unqualified"/>
<xs:attribute name="CopyAllXmlSchemaCollections" type="xs:boolean"
              use="optional" default="false" form="unqualified"/>
<xs:attribute name="LoginsList" type="xs:string"
              use="optional" default="" form="unqualified"/>
<xs:attribute name="UsersList" type="xs:string"
              use="optional" default="" form="unqualified"/>
<xs:attribute name="RulesList" type="xs:string"
              use="optional" default="" form="unqualified"/>
<xs:attribute name="TablesList" type="xs:string"
              use="optional" default="" form="unqualified"/>
<xs:attribute name="ViewsList" type="xs:string"
              use="optional" default="" form="unqualified"/>
<xs:attribute name="StoredProceduresList"
              type="xs:string" use="optional" default=""
              form="unqualified"/>
<xs:attribute name="UserDefinedFunctionsList"
              type="xs:string" use="optional" default=""
              form="unqualified"/>
<xs:attribute name="DefaultsList" type="xs:string"
              use="optional" default="" form="unqualified"/>
<xs:attribute name="UserDefinedDataTypesList"
              type="xs:string" use="optional" default=""
              form="unqualified"/>
<xs:attribute name="PartitionFunctionsList" type="xs:string"
              use="optional" default="" form="unqualified"/>
<xs:attribute name="PartitionSchemesList" type="xs:string"
              use="optional" default="" form="unqualified"/>
<xs:attribute name="SchemasList" type="xs:string"
              use="optional" default="" form="unqualified"/>
<xs:attribute name="SqlAssembliesList" type="xs:string"
              use="optional" default="" form="unqualified"/>
<xs:attribute name="UserDefinedAggregatesList"
              type="xs:string" use="optional" default=""
              form="unqualified"/>
<xs:attribute name="UserDefinedTypesList" type="xs:string"
              use="optional" default="" form="unqualified"/>
<xs:attribute name="XmlSchemaCollectionsList" type="xs:string"
              use="optional" default="" form="unqualified"/>
<xs:attribute name="DropObjectsFirst" type="xs:boolean"
              use="optional" default="false" form="unqualified"/>
<xs:attribute name="IncludeDependentObjects"
              type="xs:boolean" use="optional" default="false"
              form="unqualified"/>
<xs:attribute name="IncludeExtendedProperties" type="xs:boolean"
              use="optional" default="false" form="unqualified"/>
<xs:attribute name="CopyData" type="xs:boolean"
              use="optional" default="false" form="unqualified"/>
<xs:attribute name="ExistingData"
    type="DTS:TransferSqlServerObjectsTaskExistingDataEnum"
              use="optional" default="Replace" form="unqualified"/>
<xs:attribute name="CopySchema" type="xs:boolean"
              use="optional" default="true" form="unqualified"/>
<xs:attribute name="UseCollation" type="xs:boolean"
              use="optional" default="false" form="unqualified"/>
<xs:attribute name="CopyDatabaseUsers" type="xs:boolean"
              use="optional" default="false" form="unqualified"/>
<xs:attribute name="CopyDatabaseRoles" type="xs:boolean"
```

```
                      use="optional" default="false" form="unqualified"/>
      <xs:attribute name="CopySqlServerLogins" type="xs:boolean"
                      use="optional" default="false" form="unqualified"/>
      <xs:attribute name="CopyObjectLevelPermissions" type="xs:boolean"
                      use="optional" default="false" form="unqualified"/>
      <xs:attribute name="CopyIndexes" type="xs:boolean"
                      use="optional" default="false" form="unqualified"/>
      <xs:attribute name="CopyTriggers" type="xs:boolean"
                      use="optional" default="false" form="unqualified"/>
      <xs:attribute name="CopyFullTextIndexes" type="xs:boolean"
                      use="optional" default="false" form="unqualified"/>
      <xs:attribute name="CopyPrimaryKeys" type="xs:boolean"
                      use="optional" default="false" form="unqualified"/>
      <xs:attribute name="CopyForeignKeys" type="xs:boolean"
                      use="optional" default="false" form="unqualified"/>
      <xs:attribute name="GenerateScriptsInUnicode" type="xs:boolean"
                      use="optional" default="false" form="unqualified"/>
      <xs:attribute name="SourceTranslateChar" type="xs:boolean"
                      use="optional" default="false" form="unqualified"/>
      <xs:attribute name="DestinationTranslateChar" type="xs:boolean"
                      use="optional" default="false" form="unqualified"/>
      <xs:attribute name="CopyAllDRIObjects" type="xs:boolean"
                      use="optional" default="false" form="unqualified"/>
  </xs:complexType>

  <xs:simpleType name="TransferSqlServerObjectsTaskExistingDataEnum">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Replace"/>
      <xs:enumeration value="Append"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="WMIDRTaskDataObjectDataType">
    <xs:sequence/>
    <xs:attribute name="WQLQuerySourceType" type="DTS:SourceTypeEnum"
                    use="optional" default="DirectInput" form="unqualified"/>
    <xs:attribute name="WQLQuerySource" type="xs:string"
                    use="optional" default="" form="unqualified"/>
    <xs:attribute name="DestinationType"
                    type="DTS:DestinationTypeEnum" form="unqualified"
                    use="optional" default="FileConnection"/>
    <xs:attribute name="Destination" type="xs:string"
                    use="optional" default="" form="unqualified"/>
    <xs:attribute name="OverwriteDestination"
                    type="DTS:WMITaskDataOverwriteDestinationEnum"
                    use="optional" default="KeepOriginal" form="unqualified"/>
    <xs:attribute name="OutputType"
                    type="DTS:WMITaskDataOutputTypeEnum"
                    use="optional" default="DataTable" form="unqualified"/>
    <xs:attribute name="WMIConnectionName" type="xs:string"
                    use="optional" default="" form="unqualified"/>
  </xs:complexType>

  <xs:simpleType name="WMITaskDataOverwriteDestinationEnum">
    <xs:restriction base="xs:string">
      <xs:enumeration value="KeepOriginal"/>
      <xs:enumeration value="AppendToDestination"/>
      <xs:enumeration value="OverwriteDestination"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="WMITaskDataOutputTypeEnum">
    <xs:restriction base="xs:string">
      <xs:enumeration value="DataTable"/>
      <xs:enumeration value="PropertyNameAndValue"/>
      <xs:enumeration value="PropertyValue"/>
    </xs:restriction>
  </xs:simpleType>
```

```
<xs:complexType name="WMIEWTaskDataObjectDataType">
  <xs:sequence/>
  <xs:attribute name="WMIConnectionName" type="xs:string"
                use="optional" default="" form="unqualified"/>
  <xs:attribute name="WQLQuerySourceType" type="DTS:SourceTypeEnum"
                use="optional" default="DirectInput" form="unqualified"/>
  <xs:attribute name="WQLQuerySource" type="xs:string"
                use="optional" default="" form="unqualified"/>
  <xs:attribute name="ActionAtEvent"
                type="DTS:WMIEWTaskDataActionAtEventEnum"
                use="optional" default="LogTheEventAndFireDTSEvent"
                form="unqualified"/>
  <xs:attribute name="AfterEvent"
                type="DTS:WMIEWTaskDataAfterEventEnum"
                use="optional" default="ReturnWithSuccess"
                form="unqualified"/>
  <xs:attribute name="ActionAtTimeout"
                type="DTS:WMIEWTaskDataActionAtTimeoutEnum"
                use="optional" default="LogTimeoutAndFireDTSEvent"
                form="unqualified"/>
  <xs:attribute name="AfterTimeout"
                type="DTS:WMIEWTaskDataAfterEventEnum"
                use="optional" default="ReturnWithFailure"
                form="unqualified"/>
  <xs:attribute name="NumberOfEvents" type="xs:int"
                use="optional" default="1" form="unqualified"/>
  <xs:attribute name="Timeout" type="xs:int" form="unqualified"
                use="optional" default="0"/>
</xs:complexType>

<xs:simpleType name="WMIEWTaskDataActionAtEventEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="LogTheEventAndFireDTSEvent"/>
    <xs:enumeration value="LogTheEvent"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="WMIEWTaskDataAfterEventEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="WatchfortheEventAgain"/>
    <xs:enumeration value="ReturnWithSuccess"/>
    <xs:enumeration value="ReturnWithFailure"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="WMIEWTaskDataActionAtTimeoutEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="LogTimeoutAndFireDTSEvent"/>
    <xs:enumeration value="LogTimeout"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="XMLTaskDataObjectDataType">
  <xs:sequence>
  </xs:sequence>
  <xs:attribute name="OperationType" form="unqualified"
                type="DTS:XMLTaskOperationTypeEnum"
                use="optional" default="Diff"/>
  <xs:attribute name="SourceType" type="DTS:SourceTypeEnum"
                use="optional" default="FileConnection"
                form="unqualified"/>
  <xs:attribute name="Source" type="xs:string" form="unqualified"
                use="optional" default="" />
  <xs:attribute name="SecondOperandType" type="DTS:SourceTypeEnum"
                use="optional" default="DirectInput" form="unqualified"/>
  <xs:attribute name="SecondOperand" type="xs:string"
                use="optional" default="" form="unqualified"/>
  <xs:attribute name="SaveOperationResult" form="unqualified"
                use="optional" default="False">
    <xs:simpleType>
```

```
            <xs:restriction base="xs:string">
              <xs:enumeration value="True"/>
              <xs:enumeration value="False"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="DestinationType"
                      type="DTS:DestinationTypeEnum" form="unqualified"
                      use="optional" default="FileConnection" />
        <xs:attribute name="Destination" type="xs:string"
                      use="optional" default="" form="unqualified"/>
        <xs:attribute name="OverwriteDestination" form="unqualified"
                      use="optional" default="False">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="True"/>
              <xs:enumeration value="False"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="FailOpOnValidationFail" form="unqualified"
                      use="optional" default="False">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="True"/>
              <xs:enumeration value="False"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="FailOpOnDifference" form="unqualified"
                      use="optional" default="False">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="True"/>
              <xs:enumeration value="False"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="SaveDiffGram" form="unqualified"
                      use="optional" default="False">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="True"/>
              <xs:enumeration value="False"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="DiffGramDestinationType" form="unqualified"
                      type="DTS:DestinationTypeEnum" use="optional"
                      default="FileConnection"/>
        <xs:attribute name="DiffGramDestination" form="unqualified"
                      type="xs:string" use="optional" default=""/>
        <xs:attribute name="DiffOptions" type="DTS:XMLTaskDiffOptionsType"
                      use="optional" default="0" form="unqualified"/>
        <xs:attribute name="DiffAlgorithm" form="unqualified"
                      type="DTS:XMLTaskDiffAlgorithmEnum"
                      use="optional" default="Auto"/>
        <xs:attribute name="XPathStringSourceType"
                      type="DTS:SourceTypeEnum" form="unqualified"
                      use="optional" default="DirectInput"/>
        <xs:attribute name="XPathStringSource" type="xs:string"
                      use="optional" default="" form="unqualified"/>
        <xs:attribute name="XPathOperation" form="unqualified"
                      type="DTS:XMLTaskXPathOperationEnum"
                      use="optional" default="Evaluation" />
        <xs:attribute name="PutResultInOneNode" form="unqualified"
                      use="optional" default="False">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="True"/>
```

```
          <xs:enumeration value="False"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="ValidationType"
                  type="DTS:XMLTaskValidationTypeEnum"
                  use="optional" default="XSD" form="unqualified"/>
  </xs:complexType>

  <xs:simpleType name="XMLTaskOperationTypeEnum">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Diff"/>
      <xs:enumeration value="Validate"/>
      <xs:enumeration value="Merge"/>
      <xs:enumeration value="XSLT"/>
      <xs:enumeration value="XPATH"/>
      <xs:enumeration value="Patch"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="XMLTaskDiffOptionsType">
    <xs:restriction base="xs:int">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="255"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="XMLTaskDiffAlgorithmEnum">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Auto"/>
      <xs:enumeration value="Fast"/>
      <xs:enumeration value="Precise"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="XMLTaskXPathOperationEnum">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Evaluation"/>
      <xs:enumeration value="NodeList"/>
      <xs:enumeration value="Values"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="XMLTaskValidationTypeEnum">
    <xs:restriction base="xs:string">
      <xs:enumeration value="DTD"/>
      <xs:enumeration value="XSD"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="InnerObjectObjectDataType">
    <xs:sequence>
      <xs:any minOccurs="0" maxOccurs="unbounded"
              namespace="##any" processContents="skip"/>
    </xs:sequence>
  </xs:complexType>

  <!--Support Types Contained by ObjectData for Executables-->

  <!--pipelineObjectDataType-->

  <xs:complexType name="pipelineObjectDataType">
    <xs:sequence>
      <xs:sequence>
        <xs:element name="components" type="DTS:PipelineComponentsType"
                    form="unqualified"/>
        <xs:element name="paths" type="DTS:PipelinePathsType"
                    minOccurs="0" form="unqualified"/>
      </xs:sequence>
    </xs:sequence>
```

```
      <xs:attribute name="id" type="xs:string" form="unqualified"
                    use="required"/>
      <xs:attribute name="name" type="xs:string" form="unqualified"
                    use="optional" default=""/>
      <xs:attribute name="version" type="xs:int" form="unqualified"
                    use="optional" fixed="1"/>
      <xs:attribute name="description" type="xs:string"
                    use="optional" default="" form="unqualified"/>
      <xs:attribute name="defaultBufferMaxRows" type="xs:int"
                    use="optional" default="10000" form="unqualified"/>
      <xs:attribute name="engineThreads" type="xs:int"
                    use="optional" default="10" form="unqualified"/>
      <xs:attribute name="defaultBufferSize" type="xs:int"
                    use="optional" default="10485760" form="unqualified"/>
      <xs:attribute name="BLOBTempStoragePath" type="xs:string"
                    use="optional" default="" form="unqualified"/>
      <xs:attribute name="bufferTempStoragePath" type="xs:string"
                    use="optional" default="" form="unqualified"/>
      <xs:attribute name="runInOptimizedMode" type="xs:boolean"
                    use="optional" default="true" form="unqualified"/>
  </xs:complexType>

  <xs:complexType name="PipelineComponentsType">
    <xs:sequence>
      <xs:element name="component" type="DTS:PipelineComponentType"
                    form="unqualified" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="PipelinePathsType">
    <xs:sequence>
      <xs:element name="path" type="DTS:PipelinePathType"
          form="unqualified" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="PipelineComponentType">
    <xs:sequence>
      <xs:element name="properties" form="unqualified"
                    type="DTS:PipelineComponentPropertiesType"
                    minOccurs="0"/>
      <xs:element name="connections" form="unqualified"
                    type="DTS:PipelineComponentConnectionsType"
                    minOccurs="0"/>
      <xs:element name="inputs" form="unqualified"
                    type="DTS:PipelineComponentInputsType"
                    minOccurs="0"/>
      <xs:element name="outputs" form="unqualified"
                    type="DTS:PipelineComponentOutputsType"
                    minOccurs="0"/>
    </xs:sequence>
<!--    <xs:attribute name="id" type="xs:string" form="unqualified"/>-->
    <xs:attribute name="name" type="xs:string" form="unqualified"
                    use="optional" default=""/>
    <xs:attribute name="refId" type="xs:string" use="required"
                    form="unqualified"/>
    <xs:attribute name="componentClassID" use="required"
                    type="DTS:PipelineComponentComponentClassIDEnum"
                    form="unqualified"/>
    <xs:attribute name="description" type="xs:string"
                    form="unqualified" use="optional" default=""/>
    <xs:attribute name="localeId" type="xs:int" use="optional"
                    default="-1" form="unqualified"/>
    <xs:attribute name="usesDispositions" type="xs:boolean"
                    use="optional" default="false" form="unqualified"/>
    <xs:attribute name="validateExternalMetadata" form="unqualified"
                    use="optional" default="True">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="True"/>
```

```
          <xs:enumeration value="False"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="version" type="xs:int" use="optional"
                  default="0" form="unqualified"/>
    <xs:attribute name="pipelineVersion" type="xs:int"
                  use="optional" default="0" form="unqualified"/>
    <xs:attribute name="contactInfo" type="xs:string"
                  use="optional" default="" form="unqualified"/>
  </xs:complexType>

  <xs:complexType name="PipelineComponentPropertiesType">
    <xs:sequence>
      <xs:element name="property" form="unqualified"
                  type="DTS:PipelineComponentPropertyType"
                  minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="PipelineComponentOutputPropertiesType">
    <xs:sequence>
      <xs:element name="property" form="unqualified"
                  type="DTS:PipelineComponentOutputPropertyType"
                  minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="PipelineComponentInputPropertiesType">
    <xs:sequence>
      <xs:element name="property" form="unqualified"
                  type="DTS:PipelineComponentInputPropertyType"
                  minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="PipelineComponentConnectionsType">
    <xs:sequence>
      <xs:element name="connection" form="unqualified"
                  type="DTS:PipelineComponentConnectionType"
                  minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="PipelineComponentInputsType">
    <xs:sequence>
      <xs:element name="input" form="unqualified"
                  type="DTS:PipelineComponentInputType"
                  minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="PipelineComponentOutputsType">
    <xs:sequence>
      <xs:element name="output" form="unqualified"
                  type="DTS:PipelineComponentOutputType"
                  minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="Test">
    <xs:complexContent>
      <xs:restriction base="xs:anyType">
        <xs:sequence>
          <xs:element name="arrayElements" minOccurs="0">
            <xs:complexType>
              <xs:sequence>
                <xs:any minOccurs="0" maxOccurs="unbounded"/>
              </xs:sequence>
              <xs:anyAttribute namespace="##any" processContents="lax"/>
```

```
            </xs:complexType>
          </xs:element>
        </xs:sequence>
        <xs:anyAttribute namespace="##any" processContents="lax"/>
      </xs:restriction>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="PipelineComponentPropertyType" mixed="true">
  <xs:sequence>
    <xs:element name="arrayElements"
                type="DTS:PipelineComponentArrayElementsType"
                form="unqualified" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="name" use="required"
                type="DTS:PipelineComponentPropertyNameEnum"
                form="unqualified"/>
  <xs:attributeGroup
    ref="DTS:PipelineComponentPropertyAttributeGroup"/>
</xs:complexType>

<xs:attributeGroup
  name="PipelineComponentPropertyAttributeGroup">
  <xs:attribute name="dataType"
                type="DTS:PipelineComponentDataTypeEnum"
                form="unqualified"/>
  <xs:attribute name="name"
                type="xs:string"
                form="unqualified"/>
  <xs:attribute name="state"
                type="DTS:PipelineComponentStateEnum"
                form="unqualified" use="optional" default="default" />
  <xs:attribute name="isArray" type="xs:boolean"
                use="optional" default="false" form="unqualified"/>
  <xs:attribute name="description" type="xs:string"
                use="optional" default="" form="unqualified"/>
  <xs:attribute name="typeConverter" type="xs:string"
                use="optional" default="" form="unqualified"/>
  <xs:attribute name="UITypeEditor" type="xs:string"
                use="optional" default="" form="unqualified"/>
  <xs:attribute name="containsID" type="xs:boolean"
                use="optional" default="false" form="unqualified"/>
  <xs:attribute name="expressionType"
                type="DTS:PipelineComponentExpressionTypeEnum"
                use="optional" default="None" form="unqualified"/>
</xs:attributeGroup>

<xs:complexType name="PipelineComponentOutputPropertyType"
                mixed="true">
  <xs:sequence>
    <xs:element name="arrayElements"
                type="DTS:PipelineComponentArrayElementsType"
                form="unqualified" minOccurs="0"/>
  </xs:sequence>
  <xs:attributeGroup
    ref="DTS:PipelineComponentPropertyAttributeGroup"/>
  <xs:attribute name="name" use="required"
                type="DTS:PipelineComponentOutputPropertyNameEnum"
                form="unqualified"/>
</xs:complexType>

<xs:complexType name="PipelineComponentInputPropertyType"
                mixed="true">
  <xs:sequence>
    <xs:element name="arrayElements"
                type="DTS:PipelineComponentArrayElementsType"
                form="unqualified" minOccurs="0"/>
  </xs:sequence>
  <xs:attributeGroup
    ref="DTS:PipelineComponentPropertyAttributeGroup"/>
```

```xml
            <xs:attribute name="name" use="required"
                          type="DTS:PipelineComponentInputPropertyNameEnum"
                          form="unqualified"/>
  </xs:complexType>

  <xs:complexType name="PipelineComponentArrayElementsType">
    <xs:sequence>
      <xs:element name="arrayElement"
                  type="DTS:PipelineComponentArrayElementType"
                  form="unqualified" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="arrayElementCount" type="xs:int"
                  form="unqualified"/>
  </xs:complexType>

  <xs:complexType name="PipelineComponentArrayElementType">
    <xs:simpleContent>
      <xs:extension base="xs:anySimpleType">
        <xs:attribute name="dataType"
                      type="DTS:PipelineComponentDataTypeEnum"
                      form="unqualified"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>

  <xs:simpleType name="PipelineComponentComponentClassIDEnum">
    <xs:restriction base="DTS:uuid">
      <!--Transforms-->
      <!--Aggregate-->
      <xs:enumeration value="{5B201335-B360-485C-BB93-75C34E09B3D3}"/>
      <!--Audit-->
      <xs:enumeration value="{4FCE8297-9C65-40A9-9D60-00C5CC6BC283}"/>
      <!--Cache Transform-->
      <xs:enumeration value="{BF818E79-2C1C-410D-ADEA-B2D1A04FED01}"/>
      <!--Character Map-->
      <xs:enumeration value="{C86C867A-7D40-486A-8734-F0B6943A9DD4}"/>
      <!--Conditional Split-->
      <xs:enumeration value="{7F88F654-4E20-4D14-84F4-AF9C925D3087}"/>
      <!--Copy Column-->
      <xs:enumeration value="{D231ABC4-4754-481B-8F7A-CD4BD8D09832}"/>
      <!--Data Conversion-->
      <xs:enumeration value="{62B1106C-7DB8-4EC8-ADD6-4C664DFFC54A}"/>
      <!--Data Mining Query-->
      <xs:enumeration value="{CBDFF0CC-3758-4424-A431-3C2E103069AA}"/>
      <!--Derived Column-->
      <xs:enumeration value="{49928E82-9C4E-49F0-AABE-3812B82707EC}"/>
      <!--Export Column-->
      <xs:enumeration value="{A9823821-0085-4483-952F-36277F63FFC5}"/>
      <!--Fuzzy Grouping-->
      <xs:enumeration value="{836E0D0B-0358-45B4-B888-4ADCE9356D4E}"/>
      <!--Fuzzy Lookup-->
      <xs:enumeration value="{E4A5F949-EC93-45AB-8B36-B52936257EC2}"/>
      <!--Import Column-->
      <xs:enumeration value="{9053CAE0-140B-48E4-AEEC-4E6A26E9E436}"/>
      <!--Lookup-->
      <xs:enumeration value="{671046B0-AA63-4C9F-90E4-C06E0B710CE3}"/>
      <!--Merge-->
      <xs:enumeration value="{36E0E750-2510-4776-AA6E-17EAE84FD63E}"/>
      <!--MergeJoin-->
      <xs:enumeration value="{14D43A4F-D7BD-489D-829E-6DE35750CFE4}"/>
      <!--Multicast-->
      <xs:enumeration value="{EC139FBC-694E-490B-8EA7-35690FB0F445}"/>
      <!--OLE DB Command-->
      <xs:enumeration value="{93FFEC66-CBC8-4C7F-9C6A-CB1C17A7567D}"/>
      <!--Percentage Sampling-->
      <xs:enumeration value="{05322586-D434-4593-8BE4-8CFC766D9C8B}"/>
      <!--Pivot-->
      <xs:enumeration value="{AEB2AFAE-2201-48B1-B290-42BB8C9149DD}"/>
      <!--RowCount-->
      <xs:enumeration value="{E2697D8C-70DA-42B2-8208-A19CE3A9FE41}"/>
```

```xml
    <!--Row Sampling-->
    <xs:enumeration value="{284A868A-7C3D-498C-ABBF-028CEF6AB288}"/>
    <!--Script Component, ADO.Net source, XML Source,ADO.Net
    Destination, DataReader Destination, SQL Server Compact Destination-->
    <xs:enumeration value="{874F7595-FB5F-40FF-96AF-FBFF8250E3EF}"/>
    <!--Slowly Changing Dimension-->
    <xs:enumeration value="{25BBB0C5-369B-4303-B3DF-D0DC741DEE58}"/>
    <!--Sort-->
    <xs:enumeration value="{5B1A3FF5-D366-4D75-AD1F-F19A36FCBEDB}"/>
    <!--Term Extraction-->
    <xs:enumeration value="{056F53D1-8081-4D07-8134-1D4DAC244BB2}"/>
    <!--Term Lookup-->
    <xs:enumeration value="{ACCC729A-823D-49A7-B16D-F1999AEB1D4C}"/>
    <!--Union All-->
    <xs:enumeration value="{B594E9A8-4351-4939-891C-CFE1AB93E925}"/>
    <!--Unpivot-->
    <xs:enumeration value="{B8B094A9-8809-4E06-8874-5C55606A9FDF}"/>

    <!--Destinations-->
    <!--Excel Destination-->
    <xs:enumeration value="{1F5D5712-2FBA-4CB9-A95A-86C1F336E1DA}"/>
    <!--Flat File Destination-->
    <xs:enumeration value="{8DA75FED-1B7C-407D-B2AD-2B24209CCCA4}"/>
    <!--OLE DB Destination-->
    <xs:enumeration value="{4ADA7EAA-136C-4215-8098-D7A7C27FC0D1}"/>
    <!--Data Mining Model Training-->
    <xs:enumeration value="{3D9FFAE9-B89B-43D9-80C8-B97D2740C746}"/>
    <!--Dimension Processing-->
    <xs:enumeration value="{2C2F0891-3AAA-4865-A676-D7476FE4CE90}"/>
    <!--Partition Processing-->
    <xs:enumeration value="{DA510FB7-E3A8-4D96-9F59-55E15E67FE3D}"/>
    <!--Raw File Destination-->
    <xs:enumeration value="{04762BB6-892F-4EE6-AD46-9CEB0A7EC7A2}"/>
    <!--RecordSet Destination-->
    <xs:enumeration value="{C457FD7E-CE98-4C4B-AEFE-F3AE0044F181}"/>
    <!--SQL Server Destination-->
    <xs:enumeration value="{F452EAF3-5EF0-43F1-8067-09DDF0BC6316}"/>

    <!--Sources-->
    <!--Excel Source-->
    <xs:enumeration value="{8C084929-27D1-479F-9641-ABB7CDADF1AC}"/>
    <!--Flat File Source-->
    <xs:enumeration value="{D23FD76B-F51D-420F-BBCB-19CBF6AC1AB4}"/>
    <!--OLE DB Source-->
    <xs:enumeration value="{165A526D-D5DE-47FF-96A6-F8274C19826B}"/>
    <!--Raw File Source-->
    <xs:enumeration value="{480C7D5A-CE63-405C-B338-3C7F26560EE3}"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="PipelineComponentPropertyNameEnum">
  <xs:restriction base="xs:string">
    <!--Transforms-->
    <!--Aggregate Task-->
    <xs:enumeration value="KeyScale"/>
    <xs:enumeration value="Keys"/>
    <xs:enumeration value="CountDistinctScale"/>
    <xs:enumeration value="CountDistinctKeys"/>
    <xs:enumeration value="AutoExtendFactor"/>
    <!--Audit (none)-->
    <!--Cache Transform (none)-->
    <!--Character Map (none)-->
    <!--Conditional Split (none)-->
    <!--Copy Column (none)-->
    <!--Data Conversion (none)-->
    <!--Data Mining Query-->
    <xs:enumeration value="ObjectRef"/>
    <xs:enumeration value="QueryText"/>
    <xs:enumeration value="CatalogName"/>
    <!--Derived Column Query (none)-->
```

```xml
<!--Export Column Query (none)-->
<!--Fuzzy Grouping-->
<xs:enumeration value="MinSimilarity"/>
<xs:enumeration value="Delimiters"/>
<xs:enumeration value="MaxMemoryUsage"/>
<xs:enumeration value="Exhaustive"/>
<!--Fuzzy Lookup-->
<xs:enumeration value="MatchIndexOptions"/>
<xs:enumeration value="MaxMemoryUsage"/>
<xs:enumeration value="MatchIndexName"/>
<xs:enumeration value="ReferenceTableName"/>
<xs:enumeration value="DropExistingMatchIndex"/>
<xs:enumeration value="CopyReferenceTable"/>
<xs:enumeration value="MaxOutputMatchesPerInput"/>
<xs:enumeration value="MinSimilarity"/>
<xs:enumeration value="Delimiters"/>
<xs:enumeration value="ReferenceMetadataXml"/>
<xs:enumeration value="Exhaustive"/>
<xs:enumeration value="WarmCaches"/>
<!--Import Column (none)-->
<!--Lookup-->
<xs:enumeration value="SqlCommand"/>
<xs:enumeration value="SqlCommandParam"/>
<xs:enumeration value="ConnectionType"/>
<xs:enumeration value="CacheType"/>
<xs:enumeration value="NoMatchBehavior"/>
<xs:enumeration value="NoMatchCachePercentage"/>
<xs:enumeration value="MaxMemoryUsage"/>
<xs:enumeration value="MaxMemoryUsage64"/>
<xs:enumeration value="ReferenceMetadataXml"/>
<xs:enumeration value="ParameterMap"/>
<xs:enumeration value="DefaultCodePage"/>
<!--Merge (none)-->
<!--MergeJoin (none)-->
<xs:enumeration value="JoinType"/>
<xs:enumeration value="NumKeyColumns"/>
<xs:enumeration value="TreatNullsAsEqual"/>
<xs:enumeration value="MaxBuffersPerInput"/>
<!--Multicast (none)-->
<!--OLE DB Command-->
<xs:enumeration value="CommandTimeout"/>
<xs:enumeration value="SqlCommand"/>
<xs:enumeration value="DefaultCodePage"/>
<!--Percentage Sampling-->
<xs:enumeration value="SamplingValue"/>
<xs:enumeration value="SamplingSeed"/>
<!--Pivot (none)-->
<!--RowCount-->
<xs:enumeration value="VariableName"/>
<!--Row Sampling-->
<xs:enumeration value="SamplingValue"/>
<xs:enumeration value="SamplingSeed"/>
<!--Script Component-->
<xs:enumeration value="SourceCode"/>
<xs:enumeration value="BinaryCode"/>
<xs:enumeration value="VSTAProjectName"/>
<xs:enumeration value="ScriptLanguage"/>
<xs:enumeration value="ReadOnlyVariables"/>
<xs:enumeration value="ReadWriteVariables"/>
<xs:enumeration value="BreakpointCollection"/>
<xs:enumeration value="UserComponentTypeName"/>
<!--Slowly Changing Dimension-->
<xs:enumeration value="SqlCommand"/>
<xs:enumeration value="CurrentRowWhere"/>
<xs:enumeration value="UpdateChangingAttributeHistory"/>
<xs:enumeration value="FailOnFixedAttributeChange"/>
<xs:enumeration value="InferredMemberIndicator"/>
<xs:enumeration value="EnableInferredMember"/>
<xs:enumeration value="FailOnLookupFailure"/>
<xs:enumeration value="IncomingRowChangeType"/>
```

```xml
<xs:enumeration value="DefaultCodePage"/>
<!--Sort-->
<xs:enumeration value="EliminateDuplicates"/>
<xs:enumeration value="MaximumThreads"/>
<!--Term Extraction-->
<xs:enumeration value="NeedReferenceData"/>
<xs:enumeration value="OutTermTable"/>
<xs:enumeration value="OutTermColumn"/>
<xs:enumeration value="WordOrPhrase"/>
<xs:enumeration value="ScoreType"/>
<xs:enumeration value="FrequencyThreshold"/>
<xs:enumeration value="MaxLengthOfTerm"/>
<xs:enumeration value="IsCaseSensitive"/>
<!--Term Lookup-->
<xs:enumeration value="RefTermTable"/>
<xs:enumeration value="RefTermColumn"/>
<xs:enumeration value="IsCaseSensitive"/>
<!--UnionAll (none)-->
<!--Unpivot (none)-->

<!--Data Destinations-->
<!--ADO.Net Destination-->
<xs:enumeration value="TableOrViewName"/>
<xs:enumeration value="BatchSize"/>
<xs:enumeration value="CommandTimeout"/>
<xs:enumeration value="UserComponentTypeName"/>
<!--Excel Destination-->
<xs:enumeration value="CommandTimeout"/>
<xs:enumeration value="OpenRowset"/>
<xs:enumeration value="OpenRowsetVariable"/>
<xs:enumeration value="SqlCommand"/>
<xs:enumeration value="AccessMode"/>
<xs:enumeration value="FastLoadKeepIdentity"/>
<xs:enumeration value="FastLoadKeepNulls"/>
<xs:enumeration value="FastLoadOptions"/>
<xs:enumeration value="FastLoadMaxInsertCommitSize"/>
<!--Flat File Destination-->
<xs:enumeration value="Overwrite"/>
<xs:enumeration value="Header"/>
<!--OLE DB Destination-->
<xs:enumeration value="CommandTimeout"/>
<xs:enumeration value="OpenRowset"/>
<xs:enumeration value="OpenRowsetVariable"/>
<xs:enumeration value="SqlCommand"/>
<xs:enumeration value="DefaultCodePage"/>
<xs:enumeration value="AlwaysUseDefaultCodePage"/>
<xs:enumeration value="AccessMode"/>
<xs:enumeration value="FastLoadKeepIdentity"/>
<xs:enumeration value="FastLoadKeepNulls"/>
<xs:enumeration value="FastLoadOptions"/>
<xs:enumeration value="FastLoadMaxInsertCommitSize"/>
<!--Raw File Destination-->
<xs:enumeration value="AccessMode"/>
<xs:enumeration value="FileName"/>
<xs:enumeration value="FileNameVariable"/>
<xs:enumeration value="WriteOption"/>
<!--DataReader Destination-->
<xs:enumeration value="ReadTimeout"/>
<xs:enumeration value="DataReader"/>
<xs:enumeration value="FailOnTimeout"/>
<xs:enumeration value="UserComponentTypeName"/>
<!--Data Mining Model Training-->
<xs:enumeration value="ObjectRef"/>
<!--Dimension Processing-->
<xs:enumeration value="ObjectRef"/>
<xs:enumeration value="UseDefaultConfiguration"/>
<xs:enumeration value="KeyErrorLimit"/>
<xs:enumeration value="KeyErrorLogFile"/>
<xs:enumeration value="KeyErrorAction"/>
<xs:enumeration value="KeyErrorLimitAction"/>
```

```
                    <xs:enumeration value="KeyNotFound"/>
                    <xs:enumeration value="KeyDuplicate"/>
                    <xs:enumeration value="NullKeyConvertedToUnknown"/>
                    <xs:enumeration value="NullKeyNotAllowed"/>
                    <xs:enumeration value="ProcessType"/>
                    <!--Partition Processing-->
                    <xs:enumeration value="ObjectRef"/>
                    <xs:enumeration value="UseDefaultConfiguration"/>
                    <xs:enumeration value="KeyErrorLimit"/>
                    <xs:enumeration value="KeyErrorLogFile"/>
                    <xs:enumeration value="KeyErrorAction"/>
                    <xs:enumeration value="KeyErrorLimitAction"/>
                    <xs:enumeration value="KeyNotFound"/>
                    <xs:enumeration value="KeyDuplicate"/>
                    <xs:enumeration value="NullKeyConvertedToUnknown"/>
                    <xs:enumeration value="NullKeyNotAllowed"/>
                    <xs:enumeration value="ProcessType"/>
                    <!--RecordSet Destination-->
                    <xs:enumeration value="VariableName"/>
                    <!--SQL Server Compact Destination-->
                    <xs:enumeration value="Table Name"/>
                    <xs:enumeration value="UserComponentTypeName"/>
                    <!--SQL Server Destination-->
                    <xs:enumeration value="DefaultCodePage"/>
                    <xs:enumeration value="AlwaysUseDefaultCodePage"/>
                    <xs:enumeration value="BulkInsertTableName"/>
                    <xs:enumeration value="BulkInsertCheckConstraints"/>
                    <xs:enumeration value="BulkInsertFirstRow"/>
                    <xs:enumeration value="BulkInsertFireTriggers"/>
                    <xs:enumeration value="BulkInsertKeepIdentity"/>
                    <xs:enumeration value="BulkInsertKeepNulls"/>
                    <xs:enumeration value="BulkInsertLastRow"/>
                    <xs:enumeration value="BulkInsertMaxErrors"/>
                    <xs:enumeration value="BulkInsertOrder"/>
                    <xs:enumeration value="BulkInsertTablock"/>
                    <xs:enumeration value="Timeout"/>
                    <xs:enumeration value="MaxInsertCommitSize"/>

                    <!--Data Sources-->
                    <!--ADO.Net Source-->
                    <xs:enumeration value="SqlCommand"/>
                    <xs:enumeration value="CommandTimeout"/>
                    <xs:enumeration value="AllowImplicitStringConversion"/>
                    <xs:enumeration value="TableOrViewName"/>
                    <xs:enumeration value="AccessMode"/>
                    <xs:enumeration value="UserComponentTypeName"/>
                    <!--Excel Source-->
                    <xs:enumeration value="CommandTimeout"/>
                    <xs:enumeration value="OpenRowset"/>
                    <xs:enumeration value="OpenRowsetVariable"/>
                    <xs:enumeration value="SqlCommand"/>
                    <xs:enumeration value="SqlCommandVariable"/>
                    <xs:enumeration value="AccessMode"/>
                    <xs:enumeration value="ParameterMapping"/>
                    <!--Flat File Source-->
                    <xs:enumeration value="RetainNulls"/>
                    <xs:enumeration value="FileNameColumnName"/>
                    <!--OLEDB Source Task-->
                    <xs:enumeration value="CommandTimeout"/>
                    <xs:enumeration value="OpenRowset"/>
                    <xs:enumeration value="OpenRowsetVariable"/>
                    <xs:enumeration value="SqlCommand"/>
                    <xs:enumeration value="SqlCommandVariable"/>
                    <xs:enumeration value="DefaultCodePage"/>
                    <xs:enumeration value="AlwaysUseDefaultCodePage"/>
                    <xs:enumeration value="AccessMode"/>
                    <xs:enumeration value="ParameterMapping"/>
                    <!--Raw File Source-->
                    <xs:enumeration value="AccessMode"/>
                    <xs:enumeration value="FileName"/>
```

```
          <xs:enumeration value="FileNameVariable"/>
          <!--XML Source-->
          <xs:enumeration value="UserComponentTypeName"/>
          <xs:enumeration value="XMLData"/>
          <xs:enumeration value="XMLDataVariable"/>
          <xs:enumeration value="AccessMode"/>
          <xs:enumeration value="XMLSchemaDefinition"/>
          <xs:enumeration value="UseInlineSchema"/>

      </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="PipelineComponentOutputPropertyNameEnum">
      <xs:restriction base="xs:string">
        <!--Transforms-->
        <!--Aggregate Task-->
        <xs:enumeration value="KeyScale"/>
        <xs:enumeration value="Keys"/>
        <!--Audit (none)-->
        <!--Cache Transform (none)-->
        <!--Character Map (none)-->
        <!--Conditional Split-->
        <xs:enumeration value="Expression"/>
        <xs:enumeration value="FriendlyExpression"/>
        <xs:enumeration value="EvaluationOrder"/>
        <xs:enumeration value="IsDefaultOut"/>
        <!--Copy Column (none)-->
        <!--Data Conversion (none)-->
        <!--Data Mining Query(none)-->
        <!--Derived Column Query (none)-->
        <!--Export Column Query (none)-->
        <!--Fuzzy Grouping(none)-->
        <!--Fuzzy Lookup(none)-->
        <!--Import Column (none)-->
        <!--Lookup(none)-->
        <!--Merge (none)-->
        <!--MergeJoin (none)-->
        <!--Multicast (none)-->
        <!--OLE DB Command(none)-->
        <!--Percentage Sampling-->
        <xs:enumeration value="Selected"/>
        <!--Pivot (none)-->
        <!--RowCount (none)-->
        <!--Row Sampling-->
        <xs:enumeration value="Selected"/>
        <!--Script Component(none)-->
        <!--Slowly Changing Dimension(none)-->
        <!--Sort(none)-->
        <!--Term Extraction(none)-->
        <!--Term Lookup(none)-->
        <!--UnionAll (none)-->
        <!--Unpivot (none)-->

        <!--Data Sources(none)-->
        <!--ADO.Net Source(none)-->
        <!--Excel Source(none)-->
        <!--Flat File Source(none)-->
        <!--OLEDB Source Task(none)-->
        <!--Raw File Source(none)-->
        <!--XML Source-->
        <xs:enumeration value="RowsetID"/>
      </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="PipelineComponentInputPropertyNameEnum">
      <xs:restriction base="xs:string">
      </xs:restriction>
      <!--None of the components have defined properties for
      the Input element-->
  </xs:simpleType>
```

```
<xs:simpleType name="PipelineComponentOutputColumnPropertyNameEnum">
  <xs:restriction base="xs:string">
    <!--Transforms-->
    <!--Aggregate Task-->
    <xs:enumeration value="AggregationColumnId"/>
    <xs:enumeration value="AggregationType"/>
    <xs:enumeration value="IsBig"/>
    <xs:enumeration value="AggregationComparisonFlags"/>
    <xs:enumeration value="CountDistinctKey"/>
    <xs:enumeration value="CountDistinctScale"/>
    <!--Audit-->
    <xs:enumeration value="LineageItemSelected"/>
    <!--Cache Transform (none)-->
    <!--Character Map (none)-->
    <!--Conditional Split (none)-->
    <!--Copy Column (none)-->
    <xs:enumeration value="copyColumnId"/>
    <!--Data Conversion (none)-->
    <xs:enumeration value="SourceInputColumnLineageID"/>
    <xs:enumeration value="FastParse"/>
    <!--Data Mining Query-->
    <!--Derived Column Query (none)-->
    <xs:enumeration value="Expression"/>
    <xs:enumeration value="FriendlyExpression"/>
    <!--Export Column Query (none)-->
    <!--Fuzzy Grouping-->
    <xs:enumeration value="ColumnType"/>
    <xs:enumeration value="InputId"/>
    <!--Fuzzy Lookup-->
    <xs:enumeration value="CopyFromReferenceColumn"/>
    <xs:enumeration value="SourceInputColumnLineageId"/>
    <xs:enumeration value="ColumnType"/>
    <!--Import Column (none)-->
    <!--Lookup-->
    <xs:enumeration value="CopyFromReferenceColumn"/>
    <!--Merge (none)-->
    <!--MergeJoin-->
    <xs:enumeration value="InputColumnID"/>
    <!--Multicast (none)-->
    <!--OLE DB Command (none)-->
    <!--Percentage Sampling (none)-->
    <!--Pivot-->
    <xs:enumeration value="SourceColumn"/>
    <xs:enumeration value="PivotKeyValue"/>
    <!--RowCount (none)-->
    <!--Row Sampling-->
    <xs:enumeration value="InputColumnLineageID"/>
    <!--Script Component (none)-->
    <!--Slowly Changing Dimension (none)-->
    <!--Sort-->
    <xs:enumeration value="SortColumnId"/>
    <!--Term Extraction(none)-->
    <!--Term Lookup-->
    <xs:enumeration value="CustomLineageID"/>
    <!--UnionAll (none)-->
    <!--Unpivot (none)-->
    <xs:enumeration value="PivotKey"/>

    <!--Data Sources-->
    <!--ADO.Net Source(none)-->
    <!--Excel Source(none)-->
    <!--Flat File Source-->
    <xs:enumeration value="FastParse"/>
    <xs:enumeration value="UseBinaryFormat"/>
    <!--OLEDB Source Task(none)-->
    <!--Raw File Source(none)-->
    <!--XML Source(none)-->
  </xs:restriction>
</xs:simpleType>
```

```xml
<xs:simpleType name="PipelineComponentInputColumnPropertyNameEnum">
  <xs:restriction base="xs:string">
    <!--Transforms-->
    <!--Aggregate Task-->
    <!--Audit (none)-->
    <!--Cache Transform-->
    <xs:enumeration value="CacheColumnName"/>
    <!--Character Map-->
    <xs:enumeration value="MapFlags"/>
    <!--Conditional Split (none)-->
    <!--Copy Column (none)-->
    <!--Data Conversion (none)-->
    <!--Data Mining Query(none)-->
    <!--Derived Column Query-->
    <xs:enumeration value="Expression"/>
    <xs:enumeration value="FriendlyExpression"/>
    <!--Export Column-->
    <xs:enumeration value="FileDataColumnID"/>
    <xs:enumeration value="WriteBOM"/>
    <xs:enumeration value="ForceTruncate"/>
    <xs:enumeration value="AllowAppend"/>
    <!--Fuzzy Grouping-->
    <xs:enumeration value="ToBeCleaned"/>
    <xs:enumeration value="ExactFuzzy"/>
    <xs:enumeration value="LeadingTrailingNumeralsSignificant"/>
    <xs:enumeration value="MinSimilarity"/>
    <xs:enumeration value="FuzzyComparisonFlags"/>
    <!--Fuzzy Lookup-->
    <xs:enumeration value="JoinType"/>
    <xs:enumeration value="ExactFuzzy"/>
    <xs:enumeration value="JoinToReferenceColumn"/>
    <xs:enumeration value="FuzzyComparisonFlags"/>
    <xs:enumeration value="FuzzyComparisonFlagsEx"/>
    <!--Import Column-->
    <xs:enumeration value="FileDataColumnID"/>
    <xs:enumeration value="ExpectBOM"/>
    <!--Lookup-->
    <xs:enumeration value="JoinToReferenceColumn"/>
    <xs:enumeration value="CopyFromReferenceColumn"/>
    <!--Merge-->
    <xs:enumeration value="OutputColumnLineageID"/>
    <!--MergeJoin (none)-->
    <!--Multicast (none)-->
    <!--OLE DB Command (none)-->
    <!--Percentage Sampling (none)-->
    <!--Pivot-->
    <xs:enumeration value="PivotUsage"/>
    <!--RowCount(none)-->
    <!--Row Sampling(none)-->
    <!--Script Component(none)-->
    <!--Slowly Changing Dimension-->
    <xs:enumeration value="ColumnType"/>
    <!--Sort-->
    <xs:enumeration value="NewComparisonFlags"/>
    <xs:enumeration value="NewSortKeyPosition"/>
    <!--Term Extraction (none)-->
    <!--Term Lookup-->
    <xs:enumeration value="InputColumnType"/>
    <!--UnionAll-->
    <xs:enumeration value="OutputColumnLineageID"/>
    <!--Unpivot-->
    <xs:enumeration value="DestinationColumn"/>
    <xs:enumeration value="PivotKeyValue"/>

    <!--Data Destinations-->
    <!--ADO.Net Destination(none)-->
    <!--Excel Destination(none)-->
    <!--Flat File Destination(none)-->
    <!--OLE DB Destination(none)-->
```

```xml
          <!--Raw File Destination(none)-->
          <!--DataReader Destination(none)-->
          <!--Data Mining Model Training(none)-->
          <!--Dimension Processing(none)-->
          <!--Partition Processing(none)-->
          <!--RecordSet Destination(none)-->
          <!--SQL Server Compact Destination(none)-->
          <!--SQL Server Destination(none)-->
      </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="PipelineComponentDataTypeEnum">
    <xs:restriction base="xs:string">
      <xs:enumeration value="System.Boolean"/>
      <xs:enumeration value="System.Byte"/>
      <xs:enumeration value="System.SByte"/>
      <xs:enumeration value="System.Decimal"/>
      <xs:enumeration value="System.Double"/>
      <xs:enumeration value="System.Single"/>
      <xs:enumeration value="System.Int32"/>
      <xs:enumeration value="System.UInt32"/>
      <xs:enumeration value="System.Int64"/>
      <xs:enumeration value="System.UInt64"/>
      <xs:enumeration value="System.Int16"/>
      <xs:enumeration value="System.UInt16"/>
      <xs:enumeration value="System.String"/>
      <xs:enumeration value="System.Null"/>
      <xs:enumeration value="System.DateTime"/>
      <xs:enumeration value="System.Object"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="PipelineComponentStateEnum">
    <xs:restriction base="xs:string">
      <xs:enumeration value="cdata"/>
      <xs:enumeration value="escaped"/>
      <xs:enumeration value="default"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="PipelineComponentExpressionTypeEnum">
    <xs:restriction base="xs:string">
      <xs:enumeration value="None"/>
      <xs:enumeration value="Notify"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="PipelineComponentConnectionType">
    <xs:sequence/>
    <xs:attribute name="refId" type="xs:string" use="required"
                  form="unqualified"/>
    <xs:attribute name="name" type="xs:string" use="optional"
                    default="" form="unqualified"/>
    <xs:attribute name="description" type="xs:string"
                    use="optional" default="" form="unqualified"/>
    <xs:attribute name="connectionManagerID" type="xs:string"
                    use="optional" default="" form="unqualified"/>
  </xs:complexType>

  <xs:complexType name="PipelineComponentInputType">
    <xs:sequence>
      <xs:element name="properties"
                  type="DTS:PipelineComponentInputPropertiesType"
                  minOccurs="0" form="unqualified"/>
      <xs:element name="inputColumns"
                  type="DTS:PipelineComponentInputColumnsType"
                  minOccurs="0" form="unqualified"/>
      <xs:element name="externalMetadataColumns"
                  type="DTS:PipelineComponentMetadataColumnsType"
                  minOccurs="0" form="unqualified"/>
```

```
      </xs:sequence>
      <xs:attributeGroup
        ref="DTS:PipelineComponentInputOutputElementAttributeGroup"/>
  </xs:complexType>

  <xs:complexType name="PipelineComponentInputColumnsType">
    <xs:sequence>
      <xs:element name="inputColumn"
                  type="DTS:PipelineComponentInputColumnType"
                  minOccurs="0" maxOccurs="unbounded" form="unqualified"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="PipelineComponentInputColumnType">
    <xs:sequence>
      <xs:element name="properties"
                  type="DTS:PipelineComponentInputColumnPropertiesType"
                  form="unqualified" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attributeGroup
      ref="DTS:PipelineComponentAllColumnBaseAttributeGroup"/>
    <xs:attributeGroup
      ref="DTS:PipelineComponentIOColumnBaseAttributeGroup"/>
    <xs:attribute name="usageType"   form="unqualified"
                  use="optional" default="readOnly"
                  type="DTS:PipelineComponentUsageTypeEnum"/>
    <xs:attribute name="cachedName" type="xs:string"
               use="optional" default="" form="unqualified"/>
    <xs:attribute name="cachedSortKeyPosition" type="xs:int"
               use="optional" default="0" form="unqualified"/>
    <xs:attribute name="cachedPrecision" type="xs:int" use="optional"
                  default="0" form="unqualified"/>
    <xs:attribute name="cachedScale" type="xs:int" use="optional"
                  default="0" form="unqualified"/>
    <xs:attribute name="cachedLength" type="xs:int" use="optional"
                  default="0" form="unqualified"/>
    <xs:attribute name="cachedDataType"
                  type="DTS:PipelineComponentColumnDataTypeEnum"
                  use="optional" default="empty" form="unqualified"/>
    <xs:attribute name="cachedCodepage" type="xs:int" use="optional"
                  default="0" form="unqualified"/>
    <xs:attribute name="cachedComparisonFlags" type="xs:int"
                  form="unqualified" use="optional" default="0" />
  </xs:complexType>

  <xs:complexType name="PipelineComponentInputColumnPropertiesType">
    <xs:sequence>
      <xs:element name="property" form="unqualified"
                  type="DTS:PipelineComponentInputColumnPropertyType"
                  minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="PipelineComponentInputColumnPropertyType">
    <xs:simpleContent>
      <xs:extension base="xs:anySimpleType">
        <xs:attributeGroup
          ref="DTS:PipelineComponentPropertyAttributeGroup"/>
        <xs:attribute name="name" form="unqualified" use="required"
            type="DTS:PipelineComponentInputColumnPropertyNameEnum"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>

  <xs:complexType name="PipelineComponentOutputColumnPropertiesType">
    <xs:sequence>
      <xs:element name="property" form="unqualified"
                  type="DTS:PipelineComponentOutputColumnPropertyType"
                  minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
```

```
      </xs:complexType>

      <xs:complexType name="PipelineComponentOutputColumnPropertyType">
        <xs:simpleContent>
          <xs:extension base="xs:anySimpleType">
            <xs:attributeGroup
              ref="DTS:PipelineComponentPropertyAttributeGroup"/>
            <xs:attribute name="name" form="unqualified" use="required"
                type="DTS:PipelineComponentOutputColumnPropertyNameEnum"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>

      <xs:simpleType name="PipelineComponentUsageTypeEnum">
        <xs:restriction base="xs:string">
          <xs:enumeration value="readOnly"/>
          <xs:enumeration value="readWrite"/>
          <xs:enumeration value="Ignored"/>
        </xs:restriction>
      </xs:simpleType>

      <xs:complexType name="PipelineComponentOutputType">
        <xs:sequence>
          <xs:element name="properties"
                      type="DTS:PipelineComponentOutputPropertiesType"
                      minOccurs="0" form="unqualified"/>
          <xs:element name="outputColumns"
                      type="DTS:PipelineComponentOutputColumnsType"
                      minOccurs="0" form="unqualified"/>
          <xs:element name="externalMetadataColumns"
                      type="DTS:PipelineComponentMetadataColumnsType"
                      minOccurs="0" form="unqualified"/>
        </xs:sequence>
        <xs:attributeGroup
          ref="DTS:PipelineComponentInputOutputElementAttributeGroup"/>
        <xs:attribute name="exclusionGroup" type="xs:int"
                      use="optional" default="0" form="unqualified"/>
        <xs:attribute name="synchronousInputId" type="xs:string"
                      use="optional" default="0" form="unqualified"/>
        <xs:attribute name="deleteOutputOnPathDetached" type="xs:boolean"
                      use="optional" default="false" form="unqualified"/>
        <xs:attribute name="isErrorOut" type="xs:boolean"
                      use="optional" default="false" form="unqualified"/>
        <xs:attribute name="isSorted" type="xs:boolean" use="optional"
                      default="false" form="unqualified"/>
      </xs:complexType>

      <xs:complexType name="PipelineComponentMetadataColumnsType">
        <xs:sequence>
          <xs:element name="externalMetadataColumn"
              type="DTS:PipelineComponentMetadataColumnType"
              minOccurs="0" maxOccurs="unbounded" form="unqualified"/>
        </xs:sequence>
        <xs:attribute name="isUsed" form="unqualified">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="True"/>
              <xs:enumeration value="False"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:complexType>

      <xs:complexType name="PipelineComponentMetadataColumnType">
        <xs:sequence>
          <xs:element name="properties"
              type="DTS:PipelineComponentMetadataColumnPropertiesType"
                      minOccurs="0" form="unqualified"/>
        </xs:sequence>
        <xs:attributeGroup
```

```
            ref="DTS:PipelineComponentAllColumnBaseAttributeGroup"/>
      <xs:attributeGroup
         ref="DTS:PipelineComponentColumnExtendedAttributeGroup"/>
   </xs:complexType>

   <xs:complexType name="PipelineComponentMetadataColumnPropertiesType">
      <xs:sequence>
         <xs:element name="property"
            type="DTS:PipelineComponentMetadataColumnPropertyType"
            minOccurs="0" maxOccurs="unbounded" form="unqualified"/>
      </xs:sequence>
   </xs:complexType>

   <xs:complexType name="PipelineComponentMetadataColumnPropertyType">
      <xs:simpleContent>
         <xs:extension base="xs:anySimpleType">
            <xs:attributeGroup
               ref="DTS:PipelineComponentPropertyAttributeGroup"/>
            <xs:attribute name="name" form="unqualified" use="required"
               type="DTS:PipelineComponentMetadataColumnPropertyNameEnum"/>
         </xs:extension>
      </xs:simpleContent>
   </xs:complexType>

   <xs:simpleType name="PipelineComponentMetadataColumnPropertyNameEnum">
      <xs:restriction base="xs:string">
         <!--OLE DB Command-->
         <xs:enumeration value="DBParamInfoFlags"/>
         <!--XML Source-->
         <xs:enumeration value="CLRType"/>
      </xs:restriction>
   </xs:simpleType>

   <xs:attributeGroup
      name="PipelineComponentInputOutputElementAttributeGroup">
      <xs:attribute name="refId" type="xs:string" use="required"
                  form="unqualified"/>
      <xs:attribute name="name" type="xs:string" use="optional" default=""
                  form="unqualified"/>
      <xs:attribute name="description" type="xs:string"
                  use="optional" default="" form="unqualified"/>
      <xs:attribute name="hasSideEffects" type="xs:boolean"
                  use="optional" default="false" form="unqualified"/>
      <xs:attribute name="dangling" type="xs:boolean"
                  use="optional" default="false" form="unqualified"/>
      <xs:attribute name="errorOrTruncationOperation" type="xs:string"
                  use="optional" default="" form="unqualified"/>
      <xs:attribute name="errorRowDisposition" form="unqualified"
                  type="DTS:PipelineComponentRowDispositionEnum"
                  use="optional" default="NotUsed"/>
      <xs:attribute name="truncationRowDisposition" form="unqualified"
                  type="DTS:PipelineComponentRowDispositionEnum"
                  use="optional" default="NotUsed"/>
   </xs:attributeGroup>

   <xs:attributeGroup name="PipelineComponentAllColumnBaseAttributeGroup">
<!--      <xs:attribute name="id" type="xs:string" form="unqualified"/>-->
      <xs:attribute name="refId" type="xs:string" use="required"
                  form="unqualified"/>
      <xs:attribute name="name" type="xs:string" use="optional" default=""
                  form="unqualified"/>
      <xs:attribute name="description" type="xs:string" use="optional"
                  default="" form="unqualified"/>
      <xs:attribute name="mappedColumnId" type="xs:int" use="optional"
                  default="0" form="unqualified"/>
   </xs:attributeGroup>

   <xs:attributeGroup name="PipelineComponentIOColumnBaseAttributeGroup">
      <xs:attribute name="errorOrTruncationOperation" type="xs:string"
                  use="optional" default="" form="unqualified"/>
```

```xml
        <xs:attribute name="errorRowDisposition"
                      type="DTS:PipelineComponentRowDispositionEnum"
                      form="unqualified" use="optional" default="NotUsed"/>
        <xs:attribute name="truncationRowDisposition"
                      type="DTS:PipelineComponentRowDispositionEnum"
                      form="unqualified" use="optional" default="NotUsed"/>
        <xs:attribute name="externalMetadataColumnId" type="xs:string"
                      form="unqualified"/>
        <xs:attribute name="lineageId" type="xs:string" use="required"
                      form="unqualified"/>
    </xs:attributeGroup>

    <xs:attributeGroup name="PipelineComponentColumnExtendedAttributeGroup">
        <xs:attribute name="precision" type="xs:int" use="optional" default="0"
                      form="unqualified"/>
        <xs:attribute name="scale" type="xs:int" use="optional" default="0"
                      form="unqualified"/>
        <xs:attribute name="length" type="xs:int" use="optional" default="0"
                      form="unqualified"/>
        <xs:attribute name="dataType"
                      type="DTS:PipelineComponentColumnDataTypeEnum"
                      use="optional" default="empty" form="unqualified"/>
        <xs:attribute name="codePage" type="xs:int" use="optional" default="0"
                      form="unqualified"/>
    </xs:attributeGroup>

    <xs:complexType name="PipelineComponentOutputColumnsType">
        <xs:sequence>
            <xs:element name="outputColumn"
                        type="DTS:PipelineComponentOutputColumnType"
                    minOccurs="0" maxOccurs="unbounded" form="unqualified"/>
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="PipelineComponentOutputColumnType">
        <xs:sequence>
            <xs:element name="properties"
                        type="DTS:PipelineComponentOutputColumnPropertiesType"
                        form="unqualified" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attributeGroup
          ref="DTS:PipelineComponentAllColumnBaseAttributeGroup"/>
        <xs:attributeGroup
          ref="DTS:PipelineComponentIOColumnBaseAttributeGroup"/>
        <xs:attributeGroup
          ref="DTS:PipelineComponentColumnExtendedAttributeGroup"/>
        <xs:attribute name="sortKeyPosition" type="xs:int"
                      use="optional" default="0" form="unqualified"/>
        <xs:attribute name="comparisonFlags" type="xs:int"
                      form="unqualified" use="optional" default="0" />
        <xs:attribute name="specialFlags" form="unqualified"
                      use="optional" default="0">
            <xs:simpleType>
                <xs:restriction base="xs:int">
                    <xs:minInclusive value="0"/>
                    <xs:maxInclusive value="2"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
    </xs:complexType>

    <xs:simpleType name="PipelineComponentColumnDataTypeEnum">
        <xs:restriction base="xs:string">
            <xs:enumeration value="empty"/>
            <xs:enumeration value="i1"/>
            <xs:enumeration value="ui1"/>
            <xs:enumeration value="i2"/>
            <xs:enumeration value="i4"/>
            <xs:enumeration value="r4"/>
            <xs:enumeration value="r8"/>
```

```
        <xs:enumeration value="cy"/>
        <xs:enumeration value="date"/>
        <xs:enumeration value="bool"/>
        <xs:enumeration value="variant"/>
        <xs:enumeration value="decimal"/>
        <xs:enumeration value="ui2"/>
        <xs:enumeration value="ui4"/>
        <xs:enumeration value="i8"/>
        <xs:enumeration value="ui8"/>
        <xs:enumeration value="guid"/>
        <xs:enumeration value="bytes"/>
        <xs:enumeration value="str"/>
        <xs:enumeration value="wstr"/>
        <xs:enumeration value="numeric"/>
        <xs:enumeration value="dbDate"/>
        <xs:enumeration value="dateTime"/>
        <xs:enumeration value="dbTime"/>
        <xs:enumeration value="dbTimeStamp"/>
        <xs:enumeration value="image"/>
        <xs:enumeration value="text"/>
        <xs:enumeration value="nText"/>
        <xs:enumeration value="dbTime2"/>
        <xs:enumeration value="dbTimeStamp2"/>
        <xs:enumeration value="dbTimeStampOffset"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="PipelineComponentRowDispositionEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="NotUsed"/>
    <xs:enumeration value="IgnoreFailure"/>
    <xs:enumeration value="FailComponent"/>
    <xs:enumeration value="RedirectRow"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="PipelinePathType">
  <xs:sequence/>
  <xs:attribute name="refId" type="xs:string" use="required"
                form="unqualified"/>
  <xs:attribute name="id" type="xs:string" form="unqualified"/>
  <xs:attribute name="name" type="xs:string" form="unqualified"/>
  <xs:attribute name="description" type="xs:string"
                form="unqualified"/>
  <xs:attribute name="startId" type="xs:string" form="unqualified"/>
  <xs:attribute name="endId" type="xs:string" form="unqualified"/>
</xs:complexType>

<!--ObjectData for non-Executables-->

<xs:complexType name="LogProviderObjectDataType">
  <xs:sequence>
    <xs:element name="InnerObject" form="unqualified"
                type="DTS:InnerObjectObjectDataType"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="ForEachEnumeratorObjectDataType">
  <xs:choice>
    <xs:element name="FEIEItems" type="DTS:FEIEItemsType"
                form="unqualified"/>
    <xs:element name="ForEachFileEnumeratorProperties"
                type="DTS:ForEachFileEnumeratorPropertiesType"
                form="unqualified"/>
    <xs:element name="FEEADO" type="DTS:FEEADOType"
                form="unqualified"/>
    <xs:element name="FEESchemaRowset" type="DTS:FEESchemaRowsetType"
                form="unqualified"/>
    <xs:element name="FEEFVE" type="DTS:FEEFVEType" form="unqualified"/>
    <xs:element name="FEENODELIST" type="DTS:FEENODELISTType"
```

```
                    form="unqualified"/>
        <xs:element name="FEESMO" type="DTS:FEESMOType" form="unqualified"/>
    </xs:choice>
</xs:complexType>

<xs:complexType name="FEIEItemsType">
    <xs:sequence>
        <xs:element name="FEIEItem" type="DTS:FEIEItemType"
                    maxOccurs="unbounded" form="unqualified"/>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="FEIEItemType">
    <xs:sequence>
        <xs:element name="FEIEItemValue" maxOccurs="unbounded"
                    form="unqualified" type="DTS:FEIEItemValueType"/>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="FEIEItemValueType">
    <xs:sequence/>
    <xs:attribute name="Type" type="DTS:DtsDataTypeEnum"
                    use="required" form="unqualified"/>
    <xs:attribute name="Value" type="xs:string"
                    use="required" form="unqualified"/>
</xs:complexType>

<xs:complexType name="ForEachFileEnumeratorPropertiesType">
    <xs:sequence>
        <xs:element name="FEFEProperty" maxOccurs="unbounded" form="unqualified">
            <xs:complexType>
                <xs:sequence/>
                <xs:attribute name="Folder" form="unqualified"/>
                <xs:attribute name="FileSpec" form="unqualified"/>
                <xs:attribute name="FileNameRetrievalType" form="unqualified">
                    <xs:simpleType>
                        <xs:restriction base="xs:int">
                            <xs:minInclusive value="0"/>
                            <xs:maxInclusive value="2"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:attribute>
                <xs:attribute name="Recurse" form="unqualified">
                    <xs:simpleType>
                        <xs:restriction base="xs:int">
                            <xs:minInclusive value="-1"/>
                            <xs:maxInclusive value="0"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:attribute>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="FEEADOType">
    <xs:sequence/>
    <xs:attribute name="EnumType" use="required" form="unqualified"
                    type="DTS:FEEADOEnumType"/>
    <xs:attribute name="VarName" type="xs:string"
                    use="required" form="unqualified"/>
</xs:complexType>

<xs:simpleType name="FEEADOEnumType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="EnumerateRowsInFirstTable"/>
        <xs:enumeration value="EnumerateAllRows"/>
        <xs:enumeration value="EnumerateTables"/>
    </xs:restriction>
</xs:simpleType>
```

```
<xs:complexType name="FEESchemaRowsetType">
  <xs:sequence>
    <xs:element name="Restriction" maxOccurs="unbounded"
                form="unqualified" type="DTS:RestrictionElementType"/>
  </xs:sequence>
  <xs:attribute name="Connection" type="xs:string"
                use="required" form="unqualified"/>
  <xs:attribute name="Schema" type="DTS:uuid"
                use="required" form="unqualified"/>
</xs:complexType>

<xs:complexType name="RestrictionElementType">
  <xs:sequence/>
  <xs:attribute name="Index" type="xs:int" use="required"
                form="unqualified"/>
  <xs:attribute name="Value" type="xs:string" use="required"
                form="unqualified"/>
</xs:complexType>

<xs:complexType name="FEEFVEType">
  <xs:sequence/>
  <xs:attribute name="VariableName" type="xs:string" form="unqualified" use="required"/>
</xs:complexType>

<xs:complexType name="FEENODELISTType">
  <xs:sequence/>
  <xs:attribute name="EnumerationType"
                type="DTS:NodelistEnumerationTypeEnum"
                form="unqualified" use="required"/>
  <xs:attribute name="OuterXPathSourceType" form="unqualified"
                use="required" type="DTS:SourceTypeEnum"/>
  <xs:attribute name="OuterXPathString" type="xs:string"
                form="unqualified" use="required"/>
  <xs:attribute name="InnerElementType"
                type="DTS:NodelistInnerElementEnumerationTypeEnum"
                use="required" form="unqualified"/>
  <xs:attribute name="InnerXPathSourceType" form="unqualified"
                use="required" type="DTS:SourceTypeEnum"/>
  <xs:attribute name="InnerXPathString" type="xs:string"
                form="unqualified" use="required"/>
  <xs:attribute name="SourceType" form="unqualified"
                use="required" type="DTS:SourceTypeEnum"/>
  <xs:attribute name="SourceDocument" type="xs:string"
                use="required" form="unqualified"/>
</xs:complexType>

<xs:simpleType name="NodelistInnerElementEnumerationTypeEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Navigator"/>
    <xs:enumeration value="Node"/>
    <xs:enumeration value="NodeText"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="NodelistEnumerationTypeEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Navigator"/>
    <xs:enumeration value="Node"/>
    <xs:enumeration value="NodeText"/>
    <xs:enumeration value="ElementCollection"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="FEESMOType">
  <xs:sequence/>
  <xs:attribute name="EnumURN" type="xs:string" form="unqualified" use="required"/>
</xs:complexType>

<xs:complexType name="ConnectionManagerObjectDataType">
```

```xml
        <xs:choice>
          <xs:element name="ConnectionManager" type=
              "DTS:ConnectionManagerObjectDataConnectionManagerType"/>
          <xs:element name="MsmqConnectionManager" type=
              "DTS:ConnectionManagerObjectDataMsmqConnectionManagerType"
                    form="unqualified"/>
          <xs:element name="SMOServerConnectionManager" type=
              "DTS:ConnectionManagerObjectDataSMOServerConnectionManagerType"
                    form="unqualified"/>
          <xs:element name="SmtpConnectionManager" type=
              "DTS:ConnectionManagerObjectDataSmtpConnectionManagerType"
                    form="unqualified"/>
          <xs:element name="WmiConnectionManager" type=
              "DTS:ConnectionManagerObjectDataWmiConnectionManagerType"
                    form="unqualified"/>
        </xs:choice>
      </xs:complexType>

      <xs:complexType
        name="ConnectionManagerObjectDataConnectionManagerType">
        <xs:sequence>
          <xs:choice minOccurs="0">
            <xs:element name="FlatFileColumns" type="DTS:FlatFileColumnsType"/>
            <xs:element name="CacheColumns" type="DTS:CacheColumnsType"
                      maxOccurs="unbounded"/>
            <xs:element name="FtpConnection" type="DTS:FtpConnectionType"/>
            <xs:element name="HttpConnection" type="DTS:HttpConnectionType"/>
          </xs:choice>
        </xs:sequence>
        <xs:attributeGroup
          ref="DTS:ConnectionManagerConnectionManagerAttributeGroup"/>
      </xs:complexType>

      <xs:complexType name="FlatFileColumnsType">
        <xs:sequence>
          <xs:element name="FlatFileColumn" type="DTS:FlatFileColumnType"
                      minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>

      <xs:complexType name="FlatFileColumnType">
        <xs:sequence/>
        <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
        <xs:attributeGroup ref="DTS:FlatFileColumnAttributeGroup"/>
      </xs:complexType>

      <xs:complexType name="CacheColumnsType">
        <xs:sequence>
          <xs:element name="CacheColumn" type="DTS:CacheColumnType"
                      minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>

      <xs:complexType name="CacheColumnType">
        <xs:sequence/>
        <xs:attributeGroup ref="DTS:BasePropertyAttributeGroup"/>
        <xs:attributeGroup ref="DTS:CacheColumnAttributeGroup"/>
      </xs:complexType>

      <xs:complexType name="FtpConnectionType">
        <xs:sequence>
          <xs:element name="Property"
                      minOccurs="0" maxOccurs="unbounded">
            <xs:complexType>
              <xs:complexContent>
                <xs:extension base="DTS:PropertyElementBaseType">
                  <xs:attribute name="Name"
                      type="DTS:FtpConnectionPropertyNameEnum"
                      use="required"/>
                  <xs:attribute name="Sensitive" form="unqualified">
```

```
                <xs:simpleType>
                  <xs:restriction base="xs:int">
                    <xs:minInclusive value="0"/>
                    <xs:maxInclusive value="1"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:attribute>
              <xs:attribute name="Encrypted" form="unqualified">
                <xs:simpleType>
                  <xs:restriction base="xs:int">
                    <xs:minInclusive value="0"/>
                    <xs:maxInclusive value="1"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:attribute>
            </xs:extension>
          </xs:complexContent>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attributeGroup ref="DTS:FtpConnectionAttributeGroup"/>
</xs:complexType>

<xs:complexType name="HttpConnectionType">
  <xs:sequence>
    <xs:element name="Property"
                minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:complexContent>
          <xs:extension base="DTS:PropertyElementBaseType">
            <xs:attribute name="Name"
                type="DTS:HttpConnectionPropertyNameEnum"
                use="required"/>
            <xs:attribute name="Sensitive" form="unqualified">
              <xs:simpleType>
                <xs:restriction base="xs:int">
                  <xs:minInclusive value="0"/>
                  <xs:maxInclusive value="1"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="Encrypted" form="unqualified">
              <xs:simpleType>
                <xs:restriction base="xs:int">
                  <xs:minInclusive value="0"/>
                  <xs:maxInclusive value="1"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:attribute>
          </xs:extension>
        </xs:complexContent>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attributeGroup ref="DTS:HttpConnectionAttributeGroup"/>
</xs:complexType>

<xs:complexType
  name="ConnectionManagerObjectDataMsmqConnectionManagerType">
  <xs:sequence/>
  <xs:attribute name="ConnectionString" type="xs:string"
                form="unqualified"/>
</xs:complexType>

<xs:complexType
  name="ConnectionManagerObjectDataSMOServerConnectionManagerType">
  <xs:sequence>
    <xs:element name="PassWord" type=
                "DTS:PasswordElementType"
                form="unqualified"/>
```

```
        </xs:sequence>
        <xs:attribute name="ConnectionString" type="xs:string"
                      form="unqualified" use="required"/>
    </xs:complexType>

    <xs:complexType name="PasswordElementType">
      <xs:simpleContent>
        <xs:extension base="xs:string">
          <xs:attribute name="Sensitive" form="unqualified">
            <xs:simpleType>
              <xs:restriction base="xs:int">
                <xs:minInclusive value="0"/>
                <xs:maxInclusive value="1"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
          <xs:attribute name="Encrypted" form="unqualified">
            <xs:simpleType>
              <xs:restriction base="xs:int">
                <xs:minInclusive value="0"/>
                <xs:maxInclusive value="1"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>

    <xs:complexType
      name="ConnectionManagerObjectDataSmtpConnectionManagerType">
      <xs:sequence/>
      <xs:attribute name="ConnectionString" type="xs:string"
                    form="unqualified"/>
    </xs:complexType>

    <xs:complexType
      name="ConnectionManagerObjectDataWmiConnectionManagerType">
      <xs:sequence>
        <xs:element name="PassWord" minOccurs="0"
                    type="DTS:PasswordElementType"
                    form="unqualified"/>
      </xs:sequence>
      <xs:attribute name="ConnectionString" type="xs:string"
                    form="unqualified"/>
    </xs:complexType>

    <!--Property Element Types-->

    <xs:complexType name="PropertyElementBaseType">
      <xs:simpleContent>
        <xs:extension base="xs:anySimpleType">
          <xs:attribute name="DataType" type="DTS:DtsDataTypeEnum"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>

    <xs:complexType name="PropertyExpressionElementType">
      <xs:simpleContent>
        <xs:extension base="xs:anySimpleType">
          <xs:attribute name="Name" type="xs:string" use="required"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>

    <!--Enumeration Types-->

    <xs:simpleType name="BasePropertyNameEnum">
      <xs:restriction base="xs:string">
        <xs:enumeration value="Description"/>
        <xs:enumeration value="DTSID"/>
```

```
        <xs:enumeration value="CreationName"/>
        <xs:enumeration value="ObjectName"/>
      </xs:restriction>
  </xs:simpleType>

  <xs:attributeGroup name="BasePropertyAttributeGroup">
    <xs:attribute name="Description" type="xs:string" default=""
                  use="optional" form="qualified"/>
    <xs:attribute name="DTSID" type="DTS:uuid" use="required"
                  form="qualified"/>
    <xs:attribute name="CreationName" type="xs:string" use="required"
                  form="qualified"/>
    <xs:attribute name="ObjectName" type="xs:string" use="optional"
                  form="qualified"/>
  </xs:attributeGroup>

  <xs:simpleType name="DtsDataTypeEnum">
    <xs:restriction base="xs:int">
      <xs:enumeration value="0"/>
      <xs:enumeration value="1"/>
      <xs:enumeration value="2"/>
      <xs:enumeration value="3"/>
      <xs:enumeration value="4"/>
      <xs:enumeration value="5"/>
      <xs:enumeration value="6"/>
      <xs:enumeration value="7"/>
      <xs:enumeration value="8"/>
      <xs:enumeration value="11"/>
      <xs:enumeration value="13"/>
      <xs:enumeration value="14"/>
      <xs:enumeration value="16"/>
      <xs:enumeration value="17"/>
      <xs:enumeration value="18"/>
      <xs:enumeration value="19"/>
      <xs:enumeration value="20"/>
      <xs:enumeration value="21"/>
      <xs:enumeration value="22"/>
      <xs:enumeration value="23"/>
      <xs:enumeration value="64"/>
      <xs:enumeration value="72"/>
      <xs:enumeration value="129"/>
      <xs:enumeration value="130"/>
      <xs:enumeration value="131"/>
      <xs:enumeration value="133"/>
      <xs:enumeration value="134"/>
      <xs:enumeration value="135"/>
      <xs:enumeration value="139"/>
      <xs:enumeration value="145"/>
      <xs:enumeration value="146"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:attributeGroup name="AllExecutableAttributeGroup">
    <xs:attribute name="DisableEventHandlers" type="DTS:BooleanStringCap"
                  default="False" use="optional" form="qualified"/>
  </xs:attributeGroup>

  <xs:simpleType name="BaseExecutablePropertyNameEnum">
    <xs:union memberTypes="DTS:BasePropertyNameEnum">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="ForceExecValue"/>
          <xs:enumeration value="ExecValue"/>
          <xs:enumeration value="ForceExecutionResult"/>
          <xs:enumeration value="Disabled"/>
          <xs:enumeration value="FailPackageOnFailure"/>
          <xs:enumeration value="FailParentOnFailure"/>
          <xs:enumeration value="MaxErrorCount"/>
          <xs:enumeration value="ISOLevel"/>
          <xs:enumeration value="LocaleID"/>
```

```
            <xs:enumeration value="TransactionOption"/>
            <xs:enumeration value="DelayValidation"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:union>
   </xs:simpleType>

   <xs:attributeGroup name="BaseExecutablePropertyAttributeGroup">
     <xs:attribute name="ForceExecValue" type="DTS:BooleanStringCap"
                   default="False" use="optional" form="qualified"/>
     <xs:attribute name="ExecValue" type="xs:anySimpleType" default="0"
                   use="optional" form="qualified"/>
     <xs:attribute name="ForceExecutionResult" default="-1" use="optional"
                   type="DTS:ForceExecutionResultEnum" form="qualified"/>
     <xs:attribute name="Disabled" type="DTS:BooleanStringCap" default="False"
                   use="optional" form="qualified"/>
     <xs:attribute name="FailPackageOnFailure" type="DTS:BooleanStringCap"
                   default="False" use="optional" form="qualified"/>
     <xs:attribute name="FailParentOnFailure" type="DTS:BooleanStringCap"
                   default="False" use="optional" form="qualified"/>
     <xs:attribute name="MaxErrorCount" type="xs:int" default="1"
                   use="optional" form="qualified"/>
     <xs:attribute name="ISOLevel" type="DTS:ISOLevelEnum"
                   default="1048576" use="optional" form="qualified"/>
     <xs:attribute name="LocaleID" type="xs:int" use="optional"
                   form="qualified"/>
     <xs:attribute name="TransactionOption" type="DTS:TransactionOptionEnum"
                   default="1" use="optional" form="qualified"/>
     <xs:attribute name="DelayValidation" type="DTS:BooleanStringCap"
                   default="False" use="optional" form="qualified"/>
     <xs:attribute name="ExecValueType" type="DTS:DtsDataTypeEnum"
                   use="required" form="qualified"/>
   </xs:attributeGroup>
<!--3 ENUM follow-->
<xs:simpleType name="ForceExecutionResultEnum">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="-1"/>
    <xs:maxInclusive value="2"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="ISOLevelEnum">
  <xs:restriction base="xs:int">
    <xs:enumeration value="-1"/>
    <xs:enumeration value="16"/>
    <xs:enumeration value="1033"/>
    <xs:enumeration value="4096"/>
    <xs:enumeration value="65536"/>
    <xs:enumeration value="1048576"/>
    <xs:enumeration value="16777216"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="TransactionOptionEnum">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="2"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="ExecutableTypePackagePropertyNameEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="PackageFormatVersion"/>
  </xs:restriction>
</xs:simpleType>

<xs:attributeGroup name="ExecutableTypePackageAttributeGroup">
  <xs:attribute name="VersionComments" type="xs:string" default=""
                use="optional" form="qualified"/>
  <xs:attribute name="CreatorName" type="xs:string" use="optional"
```

```
                    form="qualified"/>
    <xs:attribute name="CreatorComputerName" type="xs:string"
                    use="optional" form="qualified"/>
    <xs:attribute name="CreationDate" type="xs:dateTime" use="optional"
                    form="qualified"/>
    <xs:attribute name="PackageType" type="DTS:PackageTypeEnum" default="0"
                    use="optional" form="qualified"/>
    <xs:attribute name="ProtectionLevel" type="DTS:ProtectionLevelEnum"
                    default="1" use="optional" form="qualified"/>
    <xs:attribute name="MaxConcurrentExecutables" type="xs:int"
                    default="-1" use="optional" form="qualified"/>
    <xs:attribute name="PackagePriorityClass" default="0" use="optional"
                    type="DTS:PackagePriorityClassEnum" form="qualified"/>
    <xs:attribute name="VersionMajor" type="xs:int" default="1"
                    use="optional" form="qualified"/>
    <xs:attribute name="VersionMinor" type="xs:int" default="0"
                    use="optional" form="qualified"/>
    <xs:attribute name="VersionBuild" type="xs:int" default="0"
                    use="optional" form="qualified"/>
    <xs:attribute name="VersionGUID" type="DTS:uuid" use="optional"
                    form="qualified"/>
    <xs:attribute name="EnableConfig" type="DTS:BooleanStringCap"
                    default="False" use="optional" form="qualified"/>
    <xs:attribute name="CheckpointFileName" type="xs:string" default=""
                    use="optional" form="qualified"/>
    <xs:attribute name="SaveCheckpoints" type="DTS:BooleanStringCap"
                    default="False" use="optional" form="qualified"/>
    <xs:attribute name="CheckpointUsage" type="DTS:CheckpointUsageEnum"
                    default="0" use="optional" form="qualified"/>
    <xs:attribute name="SuppressConfigurationWarnings" default="False"
                    type="DTS:BooleanStringCap" use="optional"
                    form="qualified"/>
    <xs:attribute name="LastModifiedProductVersion" type="xs:string"
                    use="optional" form="qualified"/>
  </xs:attributeGroup>
  <!--4 ENUM follow-->
  <xs:simpleType name="PackageTypeEnum">
    <xs:restriction base="xs:int">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="6"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="ProtectionLevelEnum">
    <xs:restriction base="xs:int">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="5"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="PackagePriorityClassEnum">
    <xs:restriction base="xs:int">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="4"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="CheckpointUsageEnum">
    <xs:restriction base="xs:int">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="2"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:attributeGroup name="AnyNonPackageExecutableAttributeGroup">
    <!--Pipeline & Tasks use these attributes-->
    <xs:attribute name="ExecutionLocation" type="xs:int" default="0"
                    use="optional" form="qualified"/>
    <xs:attribute name="ExecutionAddress" type="xs:string" default=""
                    use="optional" form="qualified"/>
```

```
        <xs:attribute name="TaskContact" type="xs:string" default=""
                      use="optional" form="qualified"/>
        <!--ForLoop uses these attributes-->
        <xs:attribute name="InitExpression" type="xs:string" default=""
                      use="optional" form="qualified"/>
        <xs:attribute name="EvalExpression" type="xs:string" use="optional"
                      form="qualified"/>
        <xs:attribute name="AssignExpression" type="xs:string" default=""
                      use="optional" form="qualified"/>
        <!--ForLoop and ForEachLoop use this attribute-->
        <xs:attribute name="MaxConcurrent" type="xs:int" default="1"
                      use="optional" form="qualified"/>
        <!--Sequence uses this attribute-->
        <xs:attribute name="LockName" type="xs:string" default=""
                      use="optional" form="qualified"/>
</xs:attributeGroup>

<xs:attributeGroup name="LogProviderAttributeGroup">
    <xs:attribute name="DelayValidation" type="DTS:BooleanStringCap"
                  default="False" use="optional" form="qualified"/>
    <xs:attribute name="ConfigString" type="xs:string" default=""
                  use="optional" form="qualified"/>
</xs:attributeGroup>

<xs:simpleType name="PackageVariablePropertyNameEnum">
    <xs:union memberTypes="DTS:BasePropertyNameEnum">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:enumeration value="PackageVariableValue"/>
                <xs:enumeration value="PackageVariable"/>
                <xs:enumeration value="Namespace"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:union>
</xs:simpleType>

<xs:simpleType name="ConfigurationPropertyNameEnum">
    <xs:union memberTypes="DTS:BasePropertyNameEnum">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:enumeration value="ConfigurationType"/>
                <xs:enumeration value="ConfigurationString"/>
                <xs:enumeration value="ConfigurationVariable"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:union>
</xs:simpleType>

<xs:attributeGroup name="ConfigurationAttributeGroup">
    <xs:attribute name="ConfigurationType" type="DTS:ConfigurationTypeEnum"
                  default="0" use="optional" form="qualified"/>
    <xs:attribute name="ConfigurationString" type="xs:string" default=""
                  use="optional" form="qualified"/>
    <xs:attribute name="ConfigurationVariable" type="xs:string" default=""
                  use="optional" form="qualified"/>
</xs:attributeGroup>
<!--1 ENUM follows-->
<xs:simpleType name="ConfigurationTypeEnum">
    <xs:restriction base="xs:int">
        <xs:minInclusive value="0"/>
        <xs:maxInclusive value="10"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="PrecedenceConstraintPropertyNameEnum">
    <xs:union memberTypes="DTS:BasePropertyNameEnum">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:enumeration value="Value"/>
                <xs:enumeration value="EvalOp"/>
```

```
            <xs:enumeration value="LogicalAnd"/>
            <xs:enumeration value="Expression"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:union>
  </xs:simpleType>


  <xs:attributeGroup name="PrecedenceConstraintAttributeGroup">
    <xs:attribute name="Value" type="DTS:PrecedenceConstraintValueEnum"
                  use="optional" default="0" form="qualified"/>
    <xs:attribute name="EvalOp" type="DTS:PrecedenceConstraintEvalOpEnum"
                  use="optional" default="1" form="qualified"/>
    <xs:attribute name="LogicalAnd" type="DTS:BooleanStringCap"
                  use="optional" default="False" form="qualified"/>
    <xs:attribute name="Expression" type="xs:string" use="optional"
                  default="" form="qualified"/>
    <xs:attribute name="From" type="xs:string" use="required"
                  form="qualified"/>
    <xs:attribute name="To" type="xs:string" use="required"
                  form="qualified"/>
  </xs:attributeGroup>
  <!--2 ENUM follow-->
  <xs:simpleType name="PrecedenceConstraintValueEnum">
    <xs:restriction base="xs:int">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="2"/>
    </xs:restriction>
  </xs:simpleType>


  <xs:simpleType name="PrecedenceConstraintEvalOpEnum">
    <xs:restriction base="xs:int">
      <xs:minInclusive value="1"/>
      <xs:maxInclusive value="4"/>
    </xs:restriction>
  </xs:simpleType>


  <xs:simpleType name="LoggingOptionsPropertyNameEnum">
    <xs:restriction base="xs:string">
      <xs:enumeration value="EventFilter"/>
      <xs:enumeration value="ColumnFilter"/>
    </xs:restriction>
  </xs:simpleType>


  <xs:attributeGroup name="LoggingOptionsAttributeGroup">
    <xs:attribute name="LoggingMode" type="DTS:LoggingModeEnum" default="0"
                  use="optional" form="qualified"/>
    <xs:attribute name="FilterKind" type="DTS:FilterKindEnum" default="1"
                  use="optional" form="qualified"/>
    <xs:attribute name="EventFilter" type="xs:string" default=""
                  use="optional" form="qualified"/>
  </xs:attributeGroup>
  <!--2 ENUM follow-->
  <xs:simpleType name="LoggingModeEnum">
    <xs:restriction base="xs:int">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="2"/>
    </xs:restriction>
  </xs:simpleType>


  <xs:simpleType name="FilterKindEnum">
    <xs:restriction base="xs:int">
      <xs:enumeration value="0"/>
      <xs:enumeration value="1"/>
    </xs:restriction>
  </xs:simpleType>


  <xs:simpleType name="ColumnEventFilterPropertyNameEnum">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Computer"/>
      <xs:enumeration value="Operator"/>
```

```xml
        <xs:enumeration value="SourceName"/>
        <xs:enumeration value="SourceID"/>
        <xs:enumeration value="ExecutionID"/>
        <xs:enumeration value="MessageText"/>
        <xs:enumeration value="DataBytes"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="ForEachVariableMappingPropertyNameEnum">
    <xs:union memberTypes="DTS:BasePropertyNameEnum">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:enumeration value="VariableName"/>
                <xs:enumeration value="ValueIndex"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:union>
</xs:simpleType>

<xs:attributeGroup name="ForEachVariableMappingAttributeGroup">
    <xs:attribute name="VariableName" type="xs:string" use="required"
                form="qualified"/>
    <xs:attribute name="ValueIndex" type="xs:int" use="required"
                form="qualified"/>
</xs:attributeGroup>

<xs:simpleType name="VariablePropertyNameEnum">
    <xs:union memberTypes="DTS:BasePropertyNameEnum">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:enumeration value="Namespace"/>
                <xs:enumeration value="Expression"/>
                <xs:enumeration value="EvaluateAsExpression"/>
                <xs:enumeration value="ReadOnly"/>
                <xs:enumeration value="RaiseChangedEvent"/>
                <xs:enumeration value="IncludeInDebugDump"/>
                <xs:enumeration value="Expression"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:union>
</xs:simpleType>

<xs:attributeGroup name="VariableAttributeGroup">
    <xs:attribute name="Namespace" type="xs:string" use="optional"
                default="" form="qualified"/>
    <xs:attribute name="Expression" type="xs:string" use="optional"
                default="" form="qualified"/>
    <xs:attribute name="EvaluateAsExpression" type="DTS:BooleanStringCap"
                use="optional" default="False" form="qualified"/>
    <xs:attribute name="ReadOnly" type="DTS:BooleanStringCap" use="optional"
                default="False" form="qualified"/>
    <xs:attribute name="RaiseChangedEvent" type="DTS:BooleanStringCap"
                use="optional" default="False" form="qualified"/>
    <xs:attribute name="IncludeInDebugDump" type="xs:int" use="optional"
                default="0" form="qualified"/>
</xs:attributeGroup>
<!--1 ENUM follows-->
<xs:simpleType name="IncludeInDebugDumpEnum">
    <xs:restriction base="xs:int">
        <xs:enumeration value="2345"/>
        <xs:enumeration value="6789"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="BooleanStringCap">
    <xs:restriction base="xs:string">
        <xs:enumeration value="True"/>
        <xs:enumeration value="False"/>
    </xs:restriction>
</xs:simpleType>
```

```
<xs:simpleType name="ConnectionManagerPropertyNameEnum">
  <xs:union memberTypes="DTS:BasePropertyNameEnum">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="DelayValidation"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:union>
</xs:simpleType>

<xs:attributeGroup name="ConnectionManagerAttributeGroup">
  <xs:attribute name="DelayValidation" type="DTS:BooleanStringCap"
                default="False" use="optional" form="qualified"/>
</xs:attributeGroup>

<xs:simpleType name="ConnectionManagerObjectDataPropertyNameEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Retain"/>
    <xs:enumeration value="ConnectionString"/>
    <xs:enumeration value="FileUsageType"/>
    <xs:enumeration value="Format"/>
    <xs:enumeration value="LocaleID"/>
    <xs:enumeration value="Unicode"/>
    <xs:enumeration value="HeaderRowsToSkip"/>
    <xs:enumeration value="HeaderRowDelimiter"/>
    <xs:enumeration value="ColumnNamesInFirstDataRow"/>
    <xs:enumeration value="RowDelimiter"/>
    <xs:enumeration value="DataRowsToSkip"/>
    <xs:enumeration value="TextQualifier"/>
    <xs:enumeration value="CodePage"/>
    <xs:enumeration value="ServerName"/>
    <xs:enumeration value="UseFile"/>
    <xs:enumeration value="UseEncryption"/>
    <xs:enumeration value="RetainData"/>
  </xs:restriction>
</xs:simpleType>

<xs:attributeGroup name="ConnectionManagerConnectionManagerAttributeGroup">
  <xs:attribute name="Retain" type="DTS:BooleanStringCap" default="False"
                use="optional" form="qualified"/>
  <xs:attribute name="ConnectionString" type="xs:string" use="required"
                form="qualified"/>
  <xs:attribute name="FileUsageType" type="DTS:FileUsageTypeEnum"
                default="0" use="optional" form="qualified"/>
  <xs:attribute name="Format" type="DTS:FormatEnum" use="optional"
                form="qualified"/>
  <xs:attribute name="LocaleID" type="xs:int" use="optional"
                form="qualified"/>
  <xs:attribute name="Unicode" type="DTS:BooleanStringCap" default="False"
                use="optional" form="qualified"/>
  <xs:attribute name="HeaderRowsToSkip" type="xs:int" default="0"
                use="optional" form="qualified"/>
  <xs:attribute name="HeaderRowDelimiter" type="xs:string" use="optional"
                form="qualified"/>
  <xs:attribute name="ColumnNamesInFirstDataRow" default="False"
                type="DTS:BooleanStringCap" use="optional" form="qualified"/>
  <xs:attribute name="RowDelimiter" type="xs:string" use="optional"
                form="qualified"/>
  <xs:attribute name="DataRowsToSkip" type="xs:int" default="0"
                use="optional" form="qualified"/>
  <xs:attribute name="TextQualifier" type="xs:string" use="optional"
                form="qualified"/>
  <xs:attribute name="CodePage" type="xs:int" use="optional"
                form="qualified"/>
  <xs:attribute name="ServerName" type="xs:string" use="optional"
                form="qualified"/>
  <xs:attribute name="UseFile" type="DTS:BooleanStringCap"
                default="False" use="optional" form="qualified"/>
  <xs:attribute name="UseEncryption" type="DTS:BooleanStringCap"
```

```
                  default="False" use="optional" form="qualified"/>
   <xs:attribute name="RetainData" type="DTS:BooleanStringCap"
                  default="True" use="optional" form="qualified"/>
</xs:attributeGroup>
<!--2 ENUM follow-->
<xs:simpleType name="FileUsageTypeEnum">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="3"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="FormatEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Delimited"/>
    <xs:enumeration value="FixedWidth"/>
    <xs:enumeration value="RaggedRight"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="FlatFileColumnPropertyNameEnum">
  <xs:union memberTypes="DTS:BasePropertyNameEnum">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="ColumnType"/>
        <xs:enumeration value="ColumnDelimiter"/>
        <xs:enumeration value="ColumnWidth"/>
        <xs:enumeration value="MaximumWidth"/>
        <xs:enumeration value="DataType"/>
        <xs:enumeration value="DataPrecision"/>
        <xs:enumeration value="DataScale"/>
        <xs:enumeration value="TextQualified"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:union>
</xs:simpleType>

<xs:attributeGroup name="FlatFileColumnAttributeGroup">
  <xs:attribute name="ColumnType" type="DTS:ColumnTypeEnum"
                  default="FixedWidth" use="optional" form="qualified"/>
  <xs:attribute name="ColumnDelimiter" type="xs:string"
                  use="required" form="qualified"/>
  <xs:attribute name="ColumnWidth" type="xs:int" default="0"
                  use="optional" form="qualified"/>
  <xs:attribute name="MaximumWidth" type="xs:int" default="0"
                  use="optional" form="qualified"/>
  <xs:attribute name="DataType" type="DTS:DtsDataTypeEnum" default="0"
                  use="optional" form="qualified"/>
  <xs:attribute name="DataPrecision" type="xs:int" default="0"
                  use="optional" form="qualified"/>
  <xs:attribute name="DataScale" type="xs:int" default="0" use="optional"
                  form="qualified"/>
  <xs:attribute name="TextQualified" type="DTS:BooleanStringCap"
                  default="False" use="optional" form="qualified"/>
</xs:attributeGroup>
<!--1 ENUM follows-->
<xs:simpleType name="ColumnTypeEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Delimited"/>
    <xs:enumeration value="FixedWidth"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="CacheColumnPropertyNameEnum">
  <xs:union memberTypes="DTS:BasePropertyNameEnum">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="DataType"/>
        <xs:enumeration value="Length"/>
        <xs:enumeration value="Precision"/>
```

```
          <xs:enumeration value="Scale"/>
          <xs:enumeration value="CodePage"/>
          <xs:enumeration value="IndexPosition"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:union>
  </xs:simpleType>

  <xs:attributeGroup name="CacheColumnAttributeGroup">
    <xs:attribute name="DataType" type="DTS:DtsDataTypeEnum" default="0"
                  use="optional" form="qualified"/>
    <xs:attribute name="Length" type="xs:int" default="0" use="optional"
                  form="qualified"/>
    <xs:attribute name="Precision" type="xs:int" default="0" use="optional"
                  form="qualified"/>
    <xs:attribute name="Scale" type="xs:int" default="0" use="optional"
                  form="qualified"/>
    <xs:attribute name="CodePage" type="xs:int" default="0" use="optional"
                  form="qualified"/>
    <xs:attribute name="IndexPosition" type="xs:int" default="0"
                  use="optional" form="qualified"/>
  </xs:attributeGroup>
  <!--0 ENUM-->

  <xs:simpleType name="FtpConnectionPropertyNameEnum">
    <xs:restriction base="xs:string">
      <xs:enumeration value="ServerPassword"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:attributeGroup name="FtpConnectionAttributeGroup">
    <xs:attribute name="ServerName" type="xs:string" use="required"
                  form="qualified"/>
    <xs:attribute name="ServerPort" type="xs:int" default="21"
                  use="optional" form="qualified"/>
    <xs:attribute name="ServerUserName" type="xs:string" default=""
                  use="optional" form="qualified"/>
    <xs:attribute name="ChunkSize" type="xs:int" default="1" use="optional"
                  form="qualified"/>
    <xs:attribute name="TimeOut" type="xs:int" default="60" use="optional"
                  form="qualified"/>
    <xs:attribute name="Retries" type="xs:int" default="5" use="optional"
                  form="qualified"/>
    <xs:attribute name="UsePassiveMode" type="DTS:BooleanStringCap"
                  default="False" use="optional" form="qualified"/>
  </xs:attributeGroup>
  <!--0 ENUM-->

  <xs:simpleType name="HttpConnectionPropertyNameEnum">
    <xs:restriction base="xs:string">
      <xs:enumeration value="ProxyPassword"/>
      <xs:enumeration value="ServerPassword"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:attributeGroup name="HttpConnectionAttributeGroup">
    <xs:attribute name="ServerURL" type="xs:string" use="required"
                  form="qualified"/>
    <xs:attribute name="UseServerCredentials" type="DTS:BooleanStringCap"
                  default="False" use="optional" form="qualified"/>
    <xs:attribute name="ServerUserName" type="xs:string" default=""
                  use="optional" form="qualified"/>
    <xs:attribute name="ServerDomain" type="xs:string" default=""
                  use="optional" form="qualified"/>
    <xs:attribute name="UseSecureConnection" type="DTS:BooleanStringCap"
                  default="False" use="optional" form="qualified"/>
    <xs:attribute name="CertificateIdentifier" type="xs:string" default=""
                  use="optional" form="qualified"/>
    <xs:attribute name="Certificate" type="xs:string" default=""
                  use="optional" form="qualified"/>
```

```xml
          <xs:attribute name="TimeOut" type="xs:int" default="60" use="optional"
                      form="qualified"/>
          <xs:attribute name="ChunkSize" type="xs:int" default="1" use="optional"
                      form="qualified"/>
          <xs:attribute name="UseProxy" type="DTS:BooleanStringCap"
                      default="False" use="optional" form="qualified"/>
          <xs:attribute name="ProxyURL" type="xs:string" default="" use="optional"
                      form="qualified"/>
          <xs:attribute name="BypassProxyOnLocal" type="DTS:BooleanStringCap"
                      default="False" use="optional" form="qualified"/>
          <xs:attribute name="UseProxyCredentials" type="DTS:BooleanStringCap"
                      default="False" use="optional" form="qualified"/>
          <xs:attribute name="ProxyUserName" type="xs:string" default=""
                      use="optional" form="qualified"/>
          <xs:attribute name="ProxyDomain" type="xs:string" default=""
                      use="optional" form="qualified"/>
          <xs:attribute name="BypassList" type="xs:string" default=""
                      use="optional" form="qualified"/>
      </xs:attributeGroup>

      <xs:simpleType name="EventHandlerPropertyNameEnum">
        <xs:union memberTypes="DTS:BaseExecutablePropertyNameEnum">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="EventID"/>
              <xs:enumeration value="EventName"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:union>
      </xs:simpleType>

      <xs:attributeGroup name="EventHandlerAttributeGroup">
        <xs:attribute name="EventID" type="xs:int" use="optional"
                      default="0" form="qualified"/>
        <xs:attribute name="EventName" type="DTS:EventNameEnum"
                      use="required" form="qualified"/>
      </xs:attributeGroup>

      <xs:simpleType name="EventNameEnum">
        <xs:restriction base="xs:string">
          <xs:enumeration value="OnError"/>
          <xs:enumeration value="OnExecStatusChanged"/>
          <xs:enumeration value="OnInformation"/>
          <xs:enumeration value="OnPostExecute"/>
          <xs:enumeration value="OnPreExecute"/>
          <xs:enumeration value="OnPreValidate"/>
          <xs:enumeration value="OnProgress"/>
          <xs:enumeration value="OnQueryCancel"/>
          <xs:enumeration value="OnTaskFailed"/>
          <xs:enumeration value="OnVariableValueChanged"/>
          <xs:enumeration value="OnWarning"/>
          <xs:enumeration value="OnCustomEvent"/>
        </xs:restriction>
      </xs:simpleType>

      <xs:simpleType name="FileSystemTaskOperationTypeEnum">
        <xs:restriction base="xs:string">
          <xs:enumeration value="CopyFile" />
          <xs:enumeration value="CreateDirectory" />
          <xs:enumeration value="DeleteDirectory" />
          <xs:enumeration value="DeleteFile" />
          <xs:enumeration value="MoveDirectory" />
          <xs:enumeration value="MoveFile" />
          <xs:enumeration value="RenameFile" />
          <xs:enumeration value="SetAttributes" />
          <xs:enumeration value="DeleteDirectoryContent" />
          <xs:enumeration value="CopyDirectory" />
        </xs:restriction>
      </xs:simpleType>
```

```
<xs:simpleType name="FileSystemTaskFileAttributesEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Archive" />
    <xs:enumeration value="Hidden" />
    <xs:enumeration value="Normal" />
    <xs:enumeration value="ReadOnly" />
    <xs:enumeration value="System" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="SourceTypeEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="DirectInput"/>
    <xs:enumeration value="FileConnection"/>
    <xs:enumeration value="Variable"/>
  </xs:restriction>
</xs:simpleType>

<!--Other Misc Types-->

<xs:simpleType name="uuid">
  <xs:restriction base="xs:string">
    <xs:pattern value="\{[0-9a-zA-Z]{8}-[0-9a-zA-Z]{4}-[0-9a-zA-Z]{4}-[0-9a-zA-Z]{4}-
        [0-9a-zA-Z]{12}\}" />
  </xs:restriction>
</xs:simpleType>


</xs:schema>
```

## 5.2   WebServiceTask XSD

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:WSTask="www.microsoft.com/sqlserver/dts/tasks/webservicetask"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           targetNamespace="www.microsoft.com/sqlserver/dts/tasks/webservicetask"
           elementFormDefault="qualified"
           attributeFormDefault="qualified"
           >

  <xs:element name="WebServiceTaskData"
              type="WSTask:WSTaskDataType"/>

  <xs:complexType name="WSTaskDataType">
    <xs:sequence>
      <xs:element name="MethodInfo"
                  type="WSTask:WebServiceTaskDataMethodInfoType"/>
    </xs:sequence>
    <xs:attribute name="ConnectionName" type="xs:string" use="optional"
                  default=""/>
    <xs:attribute name="Service" type="xs:string" use="optional"
                  default=""/>
    <xs:attribute name="WSDLFile" type="xs:string" use="optional"
                  default=""/>
    <xs:attribute name="OverwriteWSDLFile" use="optional" default="False">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="True"/>
          <xs:enumeration value="False"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="OutputType"
                  type="WSTask:WebServiceTaskDataOutputTypeEnum"
                  use="optional" default="File"/>
    <xs:attribute name="OutputLocation" type="xs:string" use="optional"
                  default=""/>
```

```
        </xs:complexType>

    <xs:complexType name="WebServiceTaskDataMethodInfoType">
      <xs:sequence>
        <xs:element name="Documentation" type="xs:string"
                     minOccurs="0" maxOccurs="1"/>
        <xs:element name="ParamInfo"
                    type="WSTask:WebServiceTaskDataParamInfoType"
                    maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="MethodName" type="xs:string"/>
      <xs:attribute name="MessageName" type="xs:string"/>
    </xs:complexType>

    <xs:complexType name="WebServiceTaskDataParamInfoType">
      <xs:sequence>
        <xs:element name="VariableValue" type="xs:string"
                    minOccurs="0" maxOccurs="1"/>
        <xs:element name="PrimitiveValue" type="xs:anySimpleType"
                    minOccurs="0" maxOccurs="1"/>
      </xs:sequence>
      <xs:attribute name="Name" type="xs:string"/>
      <xs:attribute name="Datatype" type="xs:string"/>
      <xs:attribute name="ParamType" type="WSTask:ParamTypeEnum"/>
      <xs:attribute name="SeqNumber" type="xs:int"/>
    </xs:complexType>

    <xs:simpleType name="WebServiceTaskDataOutputTypeEnum">
      <xs:restriction base="xs:string">
        <xs:enumeration value="File"/>
        <xs:enumeration value="Variable"/>
      </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="ParamTypeEnum">
      <xs:restriction base="xs:string">
        <xs:enumeration value="Variable"/>
        <xs:enumeration value="Primitive"/>
      </xs:restriction>
    </xs:simpleType>

</xs:schema>
```

## 5.3  MessageQueueTask XSD

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
xmlns:MessageQueueTask="www.microsoft.com/sqlserver/dts/tasks/messagequeuetask"
          xmlns:xs="http://www.w3.org/2001/XMLSchema"
          targetNamespace="www.microsoft.com/sqlserver/dts/tasks/messagequeuetask"
          elementFormDefault="qualified"
          attributeFormDefault="qualified"
          xmlns:DTS="www.microsoft.com/SqlServer/Dts"
          >

  <xs:import namespace="www.microsoft.com/SqlServer/Dts"
     schemaLocation="./DTSX2.xsd"  />

  <xs:element name="MessageQueueTaskData"
              type="MessageQueueTask:MessageQueueTaskType"/>

  <xs:complexType name="MessageQueueTaskType">
    <xs:sequence>
    </xs:sequence>
    <xs:attribute name="CurrentPackageID" type="DTS:uuid" use="optional"
                  default=""/>
    <xs:attribute name="TaskDescription" type="xs:string" use="optional"
                  default=""/>
```

```
      <xs:attribute name="DataFileOverWritable" use="optional"
                    default="False">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="False"/>
            <xs:enumeration value="True"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="DTSMessageLineageID" type="xs:string" use="optional"
                    default=""/>
      <xs:attribute name="DTSMessagePackageID" type="xs:string" use="optional"
                    default=""/>
      <xs:attribute name="DTSMessageVersionID" type="xs:string" use="optional"
                    default=""/>
      <xs:attribute name="ErrorIfMessageTimeOut" use="optional"
                    default="False">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="False"/>
            <xs:enumeration value="True"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="QueuePath" type="xs:string"/>
      <xs:attribute name="ReceiveMessageTimeOut" type="xs:int" use="optional"
                    default="0"/>
      <xs:attribute name="ReceiveMessageType"
                    type="MessageQueueTask:MSMQMessageTypeEnum"
                    use="optional" default="DTSMQMessageType_String"/>
      <xs:attribute name="RemoveFromQueue" use="optional" default="False">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="False"/>
            <xs:enumeration value="True"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="SaveDataFileName" type="xs:string" use="optional"
                    default=""/>
      <xs:attribute name="StringCompareType"
                    type="MessageQueueTask:MSMQStringCompareTypeEnum"
                    use="optional" default="DTSMQStringMessageCompare_None"/>
      <xs:attribute name="StringCompareValue" type="xs:string" use="optional"
                    default=""/>
      <xs:attribute name="StringMessageToVariableName" type="xs:string"
                    use="optional" default=""/>
      <xs:attribute name="ReceiveVariableMessage" type="xs:string"
                    use="optional" default=""/>
      <xs:attribute name="TaskType"
                    type="MessageQueueTask:MSMQTaskTypeEnum"
                    use="optional" default="DTSMQType_Sender"/>
      <xs:attribute name="UseEncryption" use="optional" default="False">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="False"/>
            <xs:enumeration value="True"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="EncryptionAlgorithm"
                    type="MessageQueueTask:MSMQEncryptionAlgorithmEnum"
                    use="optional" default="Rc2"/>
      <xs:attribute name="Use2000Format" use="optional" default="False">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="False"/>
            <xs:enumeration value="True"/>
          </xs:restriction>
        </xs:simpleType>
```

```
          </xs:attribute>
          <xs:attribute name="DataFileName" type="xs:string" use="optional"
                        default=""/>
          <xs:attribute name="StringMessage" type="xs:string" use="optional"
                        default=""/>
          <xs:attribute name="VariableMessage" type="xs:string" use="optional"
                        default=""/>
          <xs:attribute name="MessageType"
                        type="MessageQueueTask:MSMQMessageTypeEnum"
                        use="optional" default="DTSMQMessageType_String"/>
      </xs:complexType>

      <xs:simpleType name="MSMQStringCompareTypeEnum">
        <xs:restriction base="xs:string">
          <xs:enumeration value="DTSMQStringMessageCompare_Contains"/>
          <xs:enumeration value="DTSMQStringMessageCompare_Exact"/>
          <xs:enumeration value="DTSMQStringMessageCompare_IgnoreCase"/>
          <xs:enumeration value="DTSMQStringMessageCompare_None"/>
        </xs:restriction>
      </xs:simpleType>

      <xs:simpleType name="MSMQTaskTypeEnum">
        <xs:restriction base="xs:string">
          <xs:enumeration value="DTSMQType_Receiver"/>
          <xs:enumeration value="DTSMQType_Sender"/>
        </xs:restriction>
      </xs:simpleType>

      <xs:simpleType name="MSMQMessageTypeEnum">
        <xs:restriction base="xs:string">
          <xs:enumeration value="DTSMQMessagType_StringMessageToVariable"/>
          <xs:enumeration value="DTSMQMessageType_DataFile"/>
          <xs:enumeration value="DTSMQMessageType_String"/>
          <xs:enumeration value="DTSMQMessageType_Variables"/>
        </xs:restriction>
      </xs:simpleType>

      <xs:simpleType name="MSMQEncryptionAlgorithmEnum">
        <xs:restriction base="xs:string">
          <xs:enumeration value="Rc2"/>
          <xs:enumeration value="Rc4"/>
        </xs:restriction>
      </xs:simpleType>


  </xs:schema>
```

## 5.4 SendMailTask XSD

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
  xmlns:SendMailTask="www.microsoft.com/sqlserver/dts/tasks/sendmailtask"
          xmlns:xs="http://www.w3.org/2001/XMLSchema"
          targetNamespace="www.microsoft.com/sqlserver/dts/tasks/sendmailtask"
          elementFormDefault="qualified"
          attributeFormDefault="qualified"
          xmlns:DTS="www.microsoft.com/SqlServer/Dts"
          >

  <xs:import namespace="www.microsoft.com/SqlServer/Dts"
     schemaLocation="./DTSX2.xsd"  />


  <xs:element name="SendMailTaskData"
              type="SendMailTask:SendMailTaskType"/>

  <xs:complexType name="SendMailTaskType">
    <xs:sequence>
```

```
            <xs:element name="Attachments" minOccurs="0"
                        type="SendMailTask:AttachmentsElementType"/>
      </xs:sequence>
      <xs:attribute name="SMTPServer" type="xs:string" use="optional"
                    default=""/>
      <xs:attribute name="From" type="xs:string" use="optional" default=""/>
      <xs:attribute name="To" type="xs:string" use="optional" default=""/>
      <xs:attribute name="CC" type="xs:string" use="optional" default=""/>
      <xs:attribute name="BCC" type="xs:string" use="optional" default=""/>
      <xs:attribute name="Subject" type="xs:string" use="optional"
                    default=""/>
      <xs:attribute name="Priority"
                    type="SendMailTask:SendMailPriorityEnum"
                    use="optional" default="Normal"/>
      <xs:attribute name="MessageSourceType"
                    type="DTS:SourceTypeEnum"/>
      <xs:attribute name="MessageSource" type="xs:string"
                    use="optional" default="DirectInput"/>
  </xs:complexType>

  <xs:simpleType name="SendMailPriorityEnum">
    <xs:restriction base="xs:string">
      <xs:enumeration value="High"/>
      <xs:enumeration value="Low"/>
      <xs:enumeration value="Normal"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="AttachmentsElementType">
    <xs:sequence/>
    <xs:attribute name="FileName" type="xs:string"/>
  </xs:complexType>

</xs:schema>
```

## 5.5   ActiveXScriptTask XSD

### *Applies to DTSX2 schema for version 2012/01 (DTSX2 2012/01)*

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
xmlns:ActiveXScriptTask="www.microsoft.com/sqlserver/dts/tasks/activexscripttask"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           targetNamespace="www.microsoft.com/sqlserver/dts/tasks/activexscripttask"
           elementFormDefault="qualified"
           attributeFormDefault="qualified"
           >

  <xs:element name="ActiveXScriptTaskData"
              type="ActiveXScriptTask:ActiveXScriptTaskDataType"/>

  <xs:complexType name="ActiveXScriptTaskDataType">
    <xs:sequence/>
    <xs:attribute name="ScriptingLanguage"
                  type="ActiveXScriptTask:ScriptingLanguageEnum"/>
    <xs:attribute name="ScriptText" type="xs:string"/>
    <xs:attribute name="EntryMethod" type="xs:string"/>
  </xs:complexType>

  <xs:simpleType name="ScriptingLanguageEnum">
    <xs:restriction base="xs:string">
      <xs:enumeration value="VBScript"/>
      <xs:enumeration value="JScript.Compact"/>
      <xs:enumeration value="JScript"/>
    </xs:restriction>
  </xs:simpleType>
```

```
                </xs:schema>
```

## 5.6   Exec80PackageTask XSD

### *Applies to DTSX2 schema for version 2012/01 (DTSX2 2012/01)*

```xml
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
xmlns:Exec80PackageTask="www.microsoft.com/sqlserver/dts/tasks/exec80packagetask"
            xmlns:xs="http://www.w3.org/2001/XMLSchema"
            targetNamespace="www.microsoft.com/sqlserver/dts/tasks/exec80packagetask"
            elementFormDefault="qualified"
            attributeFormDefault="qualified"
            xmlns:DTS="www.microsoft.com/SqlServer/Dts"
            >
  <xs:import namespace="www.microsoft.com/SqlServer/Dts"
        schemaLocation="./DTSX2.xsd"  />


  <xs:element name="Exec80PackageTaskData"
              type="Exec80PackageTask:Exec80PackageTaskType"/>

  <xs:complexType name="Exec80PackageTaskType">
    <xs:sequence>
      <xs:element name="DTS2000Package" type="xs:base64Binary"
                  minOccurs="0" form="unqualified"/>
    </xs:sequence>
    <xs:attribute name="Location"
                  type="Exec80PackageTask:Exec80PackageTaskLocationEnum"/>
    <xs:attribute name="PackageName" type="xs:string"/>
    <xs:attribute name="PackageID" type="xs:string"/>
    <xs:attribute name="PackageVersionGUID" type="xs:string"/>
    <xs:attribute name="SQLServer" type="xs:string"/>
    <xs:attribute name="SQLUsername" type="xs:string"/>
    <xs:attribute name="WindowsAuthentication">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="True"/>
          <xs:enumeration value="False"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="Filename" type="xs:string"/>
    <xs:attribute name="InnerGlobalVariables" type="xs:string"/>
    <xs:attribute name="OuterGlobalVariables" type="xs:string"/>
  </xs:complexType>

  <xs:simpleType name="Exec80PackageTaskLocationEnum">
    <xs:restriction base="xs:string">
      <xs:enumeration value="SqlServer"/>
      <xs:enumeration value="EmbeddedInTask"/>
      <xs:enumeration value="StorageFile"/>
    </xs:restriction>
  </xs:simpleType>

</xs:schema>
```

## 5.7   BulkInsertTask XSD

```xml
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:BulkInsertTask="www.microsoft.com/sqlserver/dts/tasks/bulkinserttask"
            xmlns:xs="http://www.w3.org/2001/XMLSchema"
            targetNamespace="www.microsoft.com/sqlserver/dts/tasks/bulkinserttask"
            elementFormDefault="qualified"
```

```
                    attributeFormDefault="qualified"
                    >

        <xs:element name="BulkInsertTaskData"
                    type="BulkInsertTask:BulkInsertTaskDataType"/>

        <xs:complexType name="BulkInsertTaskDataType">
          <xs:sequence/>
          <xs:attribute name="BatchSize" type="xs:int" use="optional"
                        default="0"/>
          <xs:attribute name="CheckConstraints" use="optional" default="True">
            <xs:simpleType>
              <xs:restriction base="xs:string">
                <xs:enumeration value="True"/>
                <xs:enumeration value="False"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
          <xs:attribute name="CodePage" type="BulkInsertTask:CodePageType"
                        use="optional" default="RAW"/>
          <xs:attribute name="SourceConnectionName" type="xs:string"
                        use="optional" default=""/>
          <xs:attribute name="DestinationConnectionName" type="xs:string"
                        use="optional" default=""/>
          <xs:attribute name="DataFileType"
                        type="BulkInsertTask:DataFileTypeEnum"
                        use="optional" default="DTSBulkInsert_DataFileType_Char"/>
          <xs:attribute name="DestinationTableName" type="xs:string"
                        use="optional" default=""/>
          <xs:attribute name="FirstRow" type="xs:int" use="optional"
                        default="1"/>
          <xs:attribute name="LastRow" type="xs:int" use="optional"
                        default="0"/>
          <xs:attribute name="UseFormatFile" use="optional" default="False">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="True"/>
              <xs:enumeration value="False"/>
            </xs:restriction>
          </xs:simpleType>
          </xs:attribute>
          <xs:attribute name="FormatFile" type="xs:string" use="optional"
                        default=""/>
          <xs:attribute name="SortedData" type="xs:string" use="optional"
                        default=""/>
          <xs:attribute name="TableLock" use="optional" default="False">
            <xs:simpleType>
              <xs:restriction base="xs:string">
                <xs:enumeration value="True"/>
                <xs:enumeration value="False"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
          <xs:attribute name="KeepIdentity" use="optional" default="False">
            <xs:simpleType>
              <xs:restriction base="xs:string">
                <xs:enumeration value="True"/>
                <xs:enumeration value="False"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
          <xs:attribute name="KeepNulls" use="optional" default="False">
            <xs:simpleType>
              <xs:restriction base="xs:string">
                <xs:enumeration value="True"/>
                <xs:enumeration value="False"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
          <xs:attribute name="FieldTerminator"
```

```
                    type="BulkInsertTask:BulkInsertTaskTerminatorEnum"
                    use="optional" default="Tab"/>
        <xs:attribute name="RowTerminator"
                    type="BulkInsertTask:BulkInsertTaskTerminatorEnum"
                    use="optional" default="{CR}{LF}"/>
        <xs:attribute name="FireTriggers" use="optional" default="False">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="True"/>
              <xs:enumeration value="False"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="MaximumErrors" type="xs:int" use="optional"
                    default="0"/>
    </xs:complexType>

    <xs:simpleType name="DataFileTypeEnum">
      <xs:restriction base="xs:string">
        <xs:enumeration value="DTSBulkInsert_DataFileType_Char"/>
        <xs:enumeration value="DTSBulkInsert_DataFileType_Native"/>
        <xs:enumeration value="DTSBulkInsert_DataFileType_WideChar"/>
        <xs:enumeration value="DTSBulkInsert_DataFileType_WideNative"/>
      </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="CodePageType">
      <xs:union memberTypes="xs:int">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="ACP"/>
            <xs:enumeration value="OEM"/>
            <xs:enumeration value="RAW"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:union>
    </xs:simpleType>

    <xs:simpleType name="BulkInsertTaskTerminatorEnum">
      <xs:restriction base="xs:string">
        <xs:enumeration value="{CR}{LF}"/>
        <xs:enumeration value="{CR}"/>
        <xs:enumeration value="{LF}"/>
        <xs:enumeration value="Semicolon {;}"/>
        <xs:enumeration value="Comma {,}"/>
        <xs:enumeration value="Tab"/>
        <xs:enumeration value="Vertical Bar {|}"/>
      </xs:restriction>
    </xs:simpleType>
</xs:schema>
```

## 5.8  SQLTask XSD

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:SQLTask="www.microsoft.com/sqlserver/dts/tasks/sqltask"
         xmlns:xs="http://www.w3.org/2001/XMLSchema"
         targetNamespace="www.microsoft.com/sqlserver/dts/tasks/sqltask"
         elementFormDefault="qualified"
         attributeFormDefault="qualified"
         xmlns:DTS="www.microsoft.com/SqlServer/Dts"
         >

    <xs:import namespace="www.microsoft.com/SqlServer/Dts"
         schemaLocation="./DTSX2.xsd"  />

    <xs:element name="SqlTaskData"
              type="SQLTask:SqlTaskDataType"/>
```

```xml
<xs:complexType name="SqlTaskDataType">
  <xs:sequence>
    <xs:element name="SelectedDatabases"
                type="SQLTask:SelectedDatabasesType"
                minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="OperatorNotifyList"
                type="SQLTask:OperatorNotifyListType"
                minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="ParameterBinding"
                type="SQLTask:SqlTaskParameterBindingType"
                minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="ResultBinding"
                type="SQLTask:SqlTaskResultBindingType"
                minOccurs="0" maxOccurs="1"/>
    <xs:element name="BackupDestinationList"
                type="SQLTask:BackupDestinationListType"
                minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attributeGroup
    ref="SQLTask:SqlTaskBaseAttributeGroup"/>
  <xs:attributeGroup
    ref="SQLTask:MaintenanceTaskBaseAttributeGroup"/>
  <xs:attributeGroup
    ref="SQLTask:BackupTaskAttributeGroup"/>
  <xs:attributeGroup
    ref="SQLTask:CheckIntegrityTaskAttributeGroup"/>
  <xs:attributeGroup
    ref="SQLTask:AgentJobTaskAttributeGroup"/>
  <xs:attributeGroup
    ref="SQLTask:HistoryCleanupTaskAttributeGroup"/>
  <xs:attributeGroup
    ref="SQLTask:MaintenanceFileCleanupTaskAttributeGroup"/>
  <xs:attributeGroup
    ref="SQLTask:NotifyOperatorTaskAttributeGroup"/>
  <xs:attributeGroup
    ref="SQLTask:RebuildIndexTaskAttributeGroup"/>
  <xs:attributeGroup
    ref="SQLTask:ReorganizeIndexTaskAttributeGroup"/>
  <xs:attributeGroup
    ref="SQLTask:ShrinkDBTaskAttributeGroup"/>
  <xs:attributeGroup
    ref="SQLTask:UpdateStatisticsTaskAttributeGroup"/>
  <xs:attribute name="RemoveOlderThan" type="xs:int"/>
  <xs:attribute name="TimeUnitsType"
                type="SQLTask:SqlTaskTimeUnitsTypeEnum"/>
  <xs:attribute name="DatabaseSelectionType"
                type="SQLTask:SqlTaskDatabaseSelectionTypeEnum"/>
</xs:complexType>

<xs:simpleType name="SqlResultTypeEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="ResultSetType_None"/>
    <xs:enumeration value="ResultSetType_Rowset"/>
    <xs:enumeration value="ResultSetType_SingleRow"/>
    <xs:enumeration value="ResultSetType_XML"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="SelectedDatabasesType">
  <xs:sequence/>
  <xs:attribute name="DatabaseName" type="xs:string"/>
</xs:complexType>

<xs:complexType name="OperatorNotifyListType">
  <xs:sequence/>
  <xs:attribute name="OperatorNotify" type="xs:string"/>
</xs:complexType>

<xs:complexType name="SqlTaskParameterBindingType">
  <xs:sequence/>
```

```xml
          <xs:attribute name="ParameterName" type="xs:string"/>
          <xs:attribute name="DtsVariableName" type="xs:string"/>
          <xs:attribute name="ParameterDirection"
                        type="SQLTask:ParameterDirectionEnum"/>
          <xs:attribute name="DataType" type="DTS:DtsDataTypeEnum"/>
          <xs:attribute name="ParameterSize" type="xs:int"/>
        </xs:complexType>

        <xs:simpleType name="ParameterDirectionEnum">
          <xs:restriction base="xs:string">
            <xs:enumeration value="Input"/>
            <xs:enumeration value="Output"/>
            <xs:enumeration value="ReturnValue"/>
          </xs:restriction>
        </xs:simpleType>

        <xs:complexType name="SqlTaskResultBindingType">
          <xs:sequence/>
          <xs:attribute name="ResultName" type="xs:string"/>
          <xs:attribute name="DtsVariableName" type="xs:string"/>
        </xs:complexType>

        <xs:complexType name="BackupDestinationListType">
          <xs:sequence/>
          <xs:attribute name="BackupDestinationLocation"
                        type="xs:string"/>
        </xs:complexType>

        <!--Attribute Groups-->


        <xs:attributeGroup name="SqlTaskBaseAttributeGroup">
          <xs:attribute name="Connection" type="DTS:uuid" use="optional"
                        default=""/>
          <xs:attribute name="TimeOut" type="xs:int" use="optional" default="0"/>
          <xs:attribute name="IsStoredProc" use="optional" default="False">
            <xs:simpleType>
              <xs:restriction base="xs:string">
                <xs:enumeration value="True"/>
                <xs:enumeration value="False"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
          <xs:attribute name="BypassPrepare" use="optional" default="True">
            <xs:simpleType>
              <xs:restriction base="xs:string">
                <xs:enumeration value="True"/>
                <xs:enumeration value="False"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
          <xs:attribute name="SqlStmtSourceType"
                        type="DTS:SourceTypeEnum"
                        use="optional" default="DirectInput"/>
          <xs:attribute name="SqlStatementSource" type="xs:string"
                        use="optional" default=""/>
          <xs:attribute name="CodePage" type="xs:int" use="optional"/>
          <xs:attribute name="ResultType"
                        type="SQLTask:SqlResultTypeEnum"
                        use="optional" default="ResultSetType_None"/>
        </xs:attributeGroup>

        <xs:attributeGroup name="MaintenanceTaskBaseAttributeGroup">
          <xs:attribute name="ServerVersion" type="xs:int"/>
          <xs:attribute name="ExtendedLogging">
            <xs:simpleType>
              <xs:restriction base="xs:string">
                <xs:enumeration value="True"/>
                <xs:enumeration value="False"/>
              </xs:restriction>
```

```xml
          </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="LocalConnectionForLogging" type="xs:string"/>
        <xs:attribute name="TaskName" type="xs:string"/>
        <xs:attribute name="IgnoreDatabasesInNotOnlineState">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="True"/>
              <xs:enumeration value="False"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:attributeGroup>

      <xs:attributeGroup name="BackupTaskAttributeGroup">
        <xs:attribute name="BackupAction" type="SQLTask:BackupActionEnum"/>
        <xs:attribute name="BackupIsIncremental">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="True"/>
              <xs:enumeration value="False"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="BackupFileGroupsFiles" type="xs:string"/>
        <xs:attribute name="BackupDeviceType"
                      type="SQLTask:BackupDeviceTypeEnum"/>
        <xs:attribute name="BackupPhisycalDestinationType"
                      type="SQLTask:BackupDeviceTypeEnum"/>
        <xs:attribute name="BackupDestinationType"
                      type="SQLTask:DestinationTypeEnum"/>
        <xs:attribute name="BackupDestinationAutoFolderPath"
                      type="xs:string"/>
        <xs:attribute name="BackupActionForExistingBackups"
                      type="SQLTask:BackupActionForExistingBackupsEnum"/>
        <xs:attribute name="BackupCreateSubFolder">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="True"/>
              <xs:enumeration value="False"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="BackupFileExtension" type="xs:string"/>
        <xs:attribute name="BackupVerifyIntegrity">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="True"/>
              <xs:enumeration value="False"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="ExpireDate" type="xs:dateTime"/>
        <xs:attribute name="RetainDays" type="xs:int"/>
        <xs:attribute name="InDays">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="True"/>
              <xs:enumeration value="False"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="UseExpiration">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="True"/>
              <xs:enumeration value="False"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
```

```
         <xs:attribute name="BackupCompressionAction"
                       type="SQLTask:BackupCompressionActionEnum"/>
         <xs:attribute name="BackupTailLog">
           <xs:simpleType>
             <xs:restriction base="xs:string">
               <xs:enumeration value="True"/>
               <xs:enumeration value="False"/>
             </xs:restriction>
           </xs:simpleType>
         </xs:attribute>
       </xs:attributeGroup>

       <xs:simpleType name="BackupActionEnum">
         <xs:restriction base="xs:int">
           <xs:minInclusive value="0"/>
           <xs:maxInclusive value="2"/>
         </xs:restriction>
       </xs:simpleType>

       <xs:simpleType name="BackupDeviceTypeEnum">
         <xs:restriction base="xs:int">
           <xs:minInclusive value="0"/>
           <xs:maxInclusive value="4"/>
         </xs:restriction>
       </xs:simpleType>

       <xs:simpleType name="DestinationTypeEnum">
         <xs:restriction base="xs:int">
           <xs:minInclusive value="0"/>
           <xs:maxInclusive value="1"/>
         </xs:restriction>
       </xs:simpleType>

       <xs:simpleType name="BackupActionForExistingBackupsEnum">
         <xs:restriction base="xs:int">
           <xs:minInclusive value="0"/>
           <xs:maxInclusive value="1"/>
         </xs:restriction>
       </xs:simpleType>

       <xs:simpleType name="BackupCompressionActionEnum">
         <xs:restriction base="xs:int">
           <xs:minInclusive value="0"/>
           <xs:maxInclusive value="2"/>
         </xs:restriction>
       </xs:simpleType>

       <xs:attributeGroup name="CheckIntegrityTaskAttributeGroup">
         <xs:attribute name="IncludeIndexes">
           <xs:simpleType>
             <xs:restriction base="xs:string">
               <xs:enumeration value="True"/>
               <xs:enumeration value="False"/>
             </xs:restriction>
           </xs:simpleType>
         </xs:attribute>
       </xs:attributeGroup>

       <xs:simpleType name="SqlTaskDatabaseSelectionTypeEnum">
         <xs:restriction base="xs:int">
           <xs:minInclusive value="1"/>
           <xs:maxInclusive value="4"/>
         </xs:restriction>
       </xs:simpleType>

       <xs:attributeGroup name="AgentJobTaskAttributeGroup">
         <xs:attribute name="AgentJobID" type="xs:string"/>
       </xs:attributeGroup>

       <xs:attributeGroup name="HistoryCleanupTaskAttributeGroup">
```

```xml
        <xs:attribute name="RemoveBackupRestoreHistory">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="True"/>
              <xs:enumeration value="False"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="RemoveAgentHistory">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="True"/>
              <xs:enumeration value="False"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="RemoveDbMaintHistory">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="True"/>
              <xs:enumeration value="False"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:attributeGroup>

      <xs:simpleType name="SqlTaskTimeUnitsTypeEnum">
        <xs:restriction base="xs:int">
          <xs:enumeration value="0"/>
          <xs:enumeration value="1"/>
          <xs:enumeration value="2"/>
          <xs:enumeration value="3"/>
          <xs:enumeration value="5"/>
        </xs:restriction>
      </xs:simpleType>

      <xs:attributeGroup name="MaintenanceFileCleanupTaskAttributeGroup">
        <xs:attribute name="FileTypeSelected" type="xs:int"/>
        <xs:attribute name="FilePath" type="xs:string"/>
        <xs:attribute name="FolderPath" type="xs:string"/>
        <xs:attribute name="CleanSubFolders">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="True"/>
              <xs:enumeration value="False"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="FileExtension" type="xs:string"/>
        <xs:attribute name="AgeBased">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="True"/>
              <xs:enumeration value="False"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="DeleteSpecificFile">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="True"/>
              <xs:enumeration value="False"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:attributeGroup>

      <xs:simpleType name="SqlTaskFileTypeSelectedEnum">
        <xs:restriction base="xs:int">
          <xs:minInclusive value="0"/>
```

```
      <xs:maxInclusive value="1"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:attributeGroup name="NotifyOperatorTaskAttributeGroup">
    <xs:attribute name="Message" type="xs:string"/>
    <xs:attribute name="Profile" type="xs:string"/>
    <xs:attribute name="Subject" type="xs:string"/>
  </xs:attributeGroup>

  <xs:attributeGroup name="RebuildIndexTaskAttributeGroup">
    <xs:attribute name="UseOriginalAmount">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="True"/>
          <xs:enumeration value="False"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="Percentage" type="xs:int"/>
    <xs:attribute name="Sort">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="True"/>
          <xs:enumeration value="False"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="KeepOnline">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="True"/>
          <xs:enumeration value="False"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:attributeGroup>

  <xs:attributeGroup name="ReorganizeIndexTaskAttributeGroup">
    <xs:attribute name="CompactLargeObjects">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="True"/>
          <xs:enumeration value="False"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:attributeGroup>

  <xs:attributeGroup name="ShrinkDBTaskAttributeGroup">
    <xs:attribute name="DatabaseSizeLimit" type="xs:int"/>
    <xs:attribute name="DatabasePercentLimit" type="xs:int"/>
    <xs:attribute name="DatabaseReturnFreeSpace">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="True"/>
          <xs:enumeration value="False"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:attributeGroup>

  <xs:attributeGroup name="UpdateStatisticsTaskAttributeGroup">
    <xs:attribute name="UpdateStatisticsType"
                  type="SQLTask:SqlTaskUpdateStatisticsTypeEnum"/>
    <xs:attribute name="UpdateScanType"
                  type="SQLTask:SqlTaskUpdateScanTypeEnum"/>
    <xs:attribute name="UpdateSampleValue" type="xs:int"/>
  </xs:attributeGroup>
```

```xml
<xs:simpleType name="SqlTaskUpdateStatisticsTypeEnum">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="2"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="SqlTaskUpdateScanTypeEnum">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="1"/>
    <xs:maxInclusive value="3"/>
  </xs:restriction>
</xs:simpleType>

<!--The following complex types are not accessed by the formal XSD.
They are used for copy/paste into section 2.6.4.x to document behavior
of specific ExecutableType types-->

<xs:complexType name="SqlTaskDataElementSqlTaskType">
  <xs:sequence/>
  <xs:attributeGroup
    ref="SQLTask:SqlTaskBaseAttributeGroup"/>
</xs:complexType>

<xs:complexType name="SqlTaskDataElementBackupTaskType">
  <xs:sequence>
    <xs:element name="SelectedDatabases"
          type="SQLTask:SelectedDatabasesType"
          minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attributeGroup
    ref="SQLTask:SqlTaskBaseAttributeGroup"/>
  <xs:attributeGroup
    ref="SQLTask:MaintenanceTaskBaseAttributeGroup"/>
  <xs:attributeGroup
    ref="SQLTask:BackupTaskAttributeGroup"/>
  <!--Need to ck elements and other attr for backup task-->
</xs:complexType>

<xs:complexType name="SqlTaskDataElementCheckIntegrityTaskType">
  <xs:sequence>
    <xs:element name="SelectedDatabases"
          type="SQLTask:SelectedDatabasesType"
          minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="BackupDestinationList"
              type="SQLTask:BackupDestinationListType"
              minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attributeGroup
    ref="SQLTask:SqlTaskBaseAttributeGroup"/>
  <xs:attributeGroup
    ref="SQLTask:MaintenanceTaskBaseAttributeGroup"/>
  <xs:attributeGroup
    ref="SQLTask:CheckIntegrityTaskAttributeGroup"/>
  <xs:attribute name="DatabaseSelectionType"
              type="SQLTask:SqlTaskDatabaseSelectionTypeEnum"/>
</xs:complexType>

<xs:complexType name="SqlTaskDataElementAgentJobTaskType">
  <xs:sequence/>
  <xs:attributeGroup
    ref="SQLTask:SqlTaskBaseAttributeGroup"/>
  <xs:attributeGroup
    ref="SQLTask:MaintenanceTaskBaseAttributeGroup"/>
  <xs:attributeGroup
    ref="SQLTask:AgentJobTaskAttributeGroup"/>
</xs:complexType>

<xs:complexType name="SqlTaskDataElementHistoryCleanupTaskType">
  <xs:sequence/>
```

```
            <xs:attributeGroup
              ref="SQLTask:SqlTaskBaseAttributeGroup"/>
            <xs:attributeGroup
              ref="SQLTask:MaintenanceTaskBaseAttributeGroup"/>
            <xs:attributeGroup
              ref="SQLTask:HistoryCleanupTaskAttributeGroup"/>
            <xs:attribute name="RemoveOlderThan" type="xs:int"/>
            <xs:attribute name="TimeUnitsType"
                          type="SQLTask:SqlTaskTimeUnitsTypeEnum"/>
        </xs:complexType>

        <xs:complexType
          name="SqlTaskDataElementMaintenanceFileCleanupTaskType">
            <xs:sequence/>
            <xs:attributeGroup
              ref="SQLTask:SqlTaskBaseAttributeGroup"/>
            <xs:attributeGroup
              ref="SQLTask:MaintenanceTaskBaseAttributeGroup"/>
            <xs:attributeGroup
              ref="SQLTask:MaintenanceFileCleanupTaskAttributeGroup"/>
            <xs:attribute name="RemoveOlderThan" type="xs:int"/>
            <xs:attribute name="TimeUnitsType"
                          type="SQLTask:SqlTaskTimeUnitsTypeEnum"/>
        </xs:complexType>

        <xs:complexType name="SqlTaskDataElementNotifyOperatorTaskType">
            <xs:sequence>
                <xs:element name="OperatorNotifyList"
                          type="SQLTask:OperatorNotifyListType"
                          minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attributeGroup
              ref="SQLTask:SqlTaskBaseAttributeGroup"/>
            <xs:attributeGroup
              ref="SQLTask:MaintenanceTaskBaseAttributeGroup"/>
            <xs:attributeGroup
              ref="SQLTask:NotifyOperatorTaskAttributeGroup"/>
        </xs:complexType>

        <xs:complexType name="SqlTaskDataElementRebuildIndexTaskType">
            <xs:sequence>
                <xs:element name="SelectedDatabases"
                        type="SQLTask:SelectedDatabasesType"
                        minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attributeGroup
              ref="SQLTask:SqlTaskBaseAttributeGroup"/>
            <xs:attributeGroup
              ref="SQLTask:MaintenanceTaskBaseAttributeGroup"/>
            <xs:attributeGroup
              ref="SQLTask:RebuildIndexTaskAttributeGroup"/>
            <xs:attribute name="DatabaseSelectionType"
                          type="SQLTask:SqlTaskDatabaseSelectionTypeEnum"/>
        </xs:complexType>

        <xs:complexType
          name="SqlTaskDataElementReorganizeIndexTaskType">
            <xs:sequence>
                <xs:element name="SelectedDatabases"
                        type="SQLTask:SelectedDatabasesType"
                        minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attributeGroup
              ref="SQLTask:SqlTaskBaseAttributeGroup"/>
            <xs:attributeGroup
              ref="SQLTask:MaintenanceTaskBaseAttributeGroup"/>
            <xs:attributeGroup
              ref="SQLTask:ReorganizeIndexTaskAttributeGroup"/>
            <xs:attribute name="DatabaseSelectionType"
                          type="SQLTask:SqlTaskDatabaseSelectionTypeEnum"/>
```

```
      </xs:complexType>

      <xs:complexType
        name="SqlTaskDataElementShrinkDatabaseTaskType">
        <xs:sequence>
          <xs:element name="SelectedDatabases"
                type="SQLTask:SelectedDatabasesType"
                minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attributeGroup
          ref="SQLTask:SqlTaskBaseAttributeGroup"/>
        <xs:attributeGroup
          ref="SQLTask:MaintenanceTaskBaseAttributeGroup"/>
        <xs:attributeGroup
          ref="SQLTask:ShrinkDBTaskAttributeGroup"/>
        <xs:attribute name="DatabaseSelectionType"
                    type="SQLTask:SqlTaskDatabaseSelectionTypeEnum"/>
      </xs:complexType>

      <xs:complexType
        name="SqlTaskDataElementUpdateStatisticsTaskType">
        <xs:sequence>
          <xs:element name="SelectedDatabases"
                type="SQLTask:SelectedDatabasesType"
                minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attributeGroup
          ref="SQLTask:SqlTaskBaseAttributeGroup"/>
        <xs:attributeGroup
          ref="SQLTask:MaintenanceTaskBaseAttributeGroup"/>
        <xs:attributeGroup
          ref="SQLTask:UpdateStatisticsTaskAttributeGroup"/>
        <xs:attribute name="DatabaseSelectionType"
                    type="SQLTask:SqlTaskDatabaseSelectionTypeEnum"/>
      </xs:complexType>

      <xs:complexType name="SqlTaskDataElementTSQLExecuteTaskType">
        <xs:sequence/>
        <xs:attributeGroup
          ref="SQLTask:SqlTaskBaseAttributeGroup"/>
        <xs:attributeGroup
          ref="SQLTask:MaintenanceTaskBaseAttributeGroup"/>
      </xs:complexType>

  </xs:schema>
```

# 6   Appendix B: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include updates to those products.

- Microsoft SQL Server 2012

- Microsoft SQL Server 2014

- Microsoft SQL Server 2016

- Microsoft SQL Server 2017

- Microsoft SQL Server 2019

- Microsoft SQL Server 2022

Exceptions, if any, are noted in this section. If an update version, service pack or Knowledge Base (KB) number appears with a product name, the behavior changed in that update. The new behavior also applies to subsequent updates unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms "SHOULD" or "SHOULD NOT" implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term "MAY" implies that the product does not follow the prescription.

<1> Section 1.6: The following table identifies DTSX2 schema versions and the products to which they are applicable. Newer releases of Microsoft SQL Server Integration Services (SSIS) support previous DTSX2 schema versions. For example, SQL Server 2019 Integration Services supports all previous DTSX2 schema versions.

| DTSX2 schema version | Applicable product |
|---|---|
| DTSX2 2012/01 | SQL Server 2012 Integration Services<br>SQL Server 2014 Integration Services |
| DTSX2 2014/01 | SQL Server 2014 Integration Services<br>SQL Server 2016 Integration Services<br>SQL Server 2017 Integration Services<br>SQL Server 2019 Integration Services<br>SQL Server 2022 Integration Services |

<2> Section 2: Integration Services provides user options for encryption. The default is that sensitive properties are not included in the clear text file. Options are provided to include sensitive information in the structure file in encrypted form.

<3> Section 2.1.2: The Integration Services system elements can appear in any order within the <xs:sequence> tags. The Integration Services UI generates package files that do not adhere to the sequence that is stated in the XSD.

<4> Section 2.4: The Integration Services user interface generates a unique value for the **refId** attribute value at the time of object creation.

<5> Section 2.4.3.1.2: In Microsoft implementations, the **InnerObject** element is always empty.

<6> Section 2.4.4.1: The Integration Services user interface generates a unique value for the **refId** attribute value at the time of object creation.

<7> Section 2.4.4.1.2.1.1: In Microsoft implementations, if a value for the **LocaleID** attribute is not specified, the default value that is set in the Microsoft Windows operating system is obtained from the system and used.

<8> Section 2.4.4.1.2.1.1: The value of the **RowDelimiter** attribute is not used by Integration Services.

<9> Section 2.4.4.1.2.1.1: In Microsoft implementations, if a value for the **CodePage** attribute is not specified, the default value that is set in the Windows operating system is obtained from the operating system and used.

<10> Section 2.4.4.1.2.1.1: The **RetainData** enumeration value is not used by Integration Services.

<11> Section 2.4.4.1.2.3: The SMO object model is available only on the Microsoft SQL Server database.

<12> Section 2.4.6: The value of this property is not processed by Integration Services.

<13> Section 2.4.6: The value of this property is not processed by Integration Services. By default, the Integration Services UI places the domain-qualified logon name of the package creator in this field.

<14> Section 2.4.6: The value of this property is not processed by Integration Services. By default, the Integration Services UI places the domain name of the computer on which the package was created in this field.

<15> Section 2.4.6: The value of this property is not processed by Integration Services. By default, the Integration Services UI places the date/time value obtained from the operating system in this field.

<16> Section 2.4.6: This field is for documentation and is not processed by Integration Services.

<17> Section 2.4.6: The **VersionBuild** value is incremented every time the package is saved when operating in the Integration Services user interface or through the Integration Services run-time object model.

<18> Section 2.4.6: In Microsoft implementations, the **VersionGUID** value changes every time the package is saved.

<19> Section 2.4.6: The Integration Services user interface sets the **LastModifiedProductVersion** attribute to the version of the installed version of SQL Server that is present on the local machine.

<20> Section 2.5.1: This integer value is not currently used by the Integration Services system.

<21> Section 2.5.1: The Integration Services user interface generates a unique value for the **refId** attribute value at the time of object creation.

<22> Section 2.5.1.3: Integration Services sets the **ExecutionLocation** value to 0.

<23> Section 2.5.1.3: Integration Services sets the **ExecutionAddress** value to EMPTY.

<24> Section 2.5.1.3: Integration Services sets **MaxConcurrent** to 1.

<25> Section 2.5.1.3: The **LockName** property is left blank by Integration Services. If a value is present, it is ignored by Integration Services.

<26> Section 2.7.1.1: In Microsoft implementations, the default for the **defaultBufferMaxRows** attribute value is 10,000 rows.

<27> Section 2.7.1.1: In Microsoft implementations, the default value for the **engineThreads** attribute is 5. The data flow engine can, in some circumstances, exceed the **engineThreads** value. If the value is set higher than the data flow engine needs, the data flow engine uses only the number of threads that it needs.

<28> Section 2.7.1.1: In Microsoft implementations, the default for the **defaultBufferSize** attribute is 10 megabytes.

<29> Section 2.7.1.1: In Microsoft implementations, the default value that is used for the **BLOBTempStoragePath** attribute value is the folder that the Windows TEMP environment variable points to.

<30> Section 2.7.1.1: In Microsoft implementations, the default value that is used for the **bufferTempStoragePath** attribute value is the folder that the Windows TEMP environment variable points to.

<31> Section 2.7.1.1.1.1: The Integration Services UI generates a unique value for the **refId** attribute value at the time of object creation.

<32> Section 2.7.1.1.1.1: In Microsoft implementations, the **validateExternalMetadata** attribute specifies whether the Microsoft Visual Studio UI validates column data against its external data source at design time.

- TRUE specifies that the user interface validates column data against external data sources at design time.

- FALSE specifies that the user interface does not validate external metadata against its external data source at design time.

<33> Section 2.7.1.1.1.1: Integration Services does not process the contents of the **contactInfo** attribute value.

<34> Section 2.7.1.1.1.1.1.1.2: In Microsoft implementations, a value of 0 for the **MaxMemoryUsage** property element value indicates that the system dynamically sets the memory usage.

<35> Section 2.7.1.1.1.1.1.1.2: In Microsoft implementations, a value of 0 for the **MaxMemoryUsage** attribute specifies that the cache size has no limit.

<36> Section 2.7.1.1.1.1.1.1.2: In Microsoft implementations, this value has to be a valid SQL Server identifier name.

<37> Section 2.7.1.1.1.1.1.1.2: The **ReferenceMetadataXml** attribute value is not used by Integration Services.

<38> Section 2.7.1.1.1.1.1.1.2: The **ReferenceMetadataXml** attribute value is not used by Integration Services.

<39> Section 2.7.1.1.1.1.1.1.2: In Microsoft implementations, the **VSTAProjectName** attribute is the project name of the Microsoft Visual Studio Tools for Applications project.

<40> Section 2.7.1.1.1.1.1.1.2: In Microsoft implementations, a value of 0 for the **MaximumThreads** attribute specifies that the maximum number of threads is unlimited.

<41> Section 2.7.1.1.1.1.1.1.2: In Microsoft implementations, the value of 0 for the **BatchSize** attribute indicates that the batch size is set to the internal buffer size.

<42> Section 2.7.1.1.1.1.1.1.2: In Microsoft implementations, the value of 0 for the **CommandTimeout** attribute indicates that there is no timeout.

<43> Section 2.7.1.1.1.1.1.1.2: In Microsoft implementations, the value of 0 for the **CommandTimeout** attribute indicates that there is no timeout.

<44> Section 2.7.1.1.1.1.1.1.2: The **FastLoadOptions** property applies to Microsoft products such as SQL Server and Microsoft Office Excel. It might not be applicable to servers or to applications from other vendors.

<45> Section 2.7.1.1.1.1.1.1.2: In Microsoft implementations, the value of 0 for the **CommandTimeout** attribute indicates that there is no timeout.

<46> Section 2.7.1.1.1.1.1.1.2: The **FastLoadOptions** property applies to Microsoft products such as SQL Server and Office Excel. It might not be applicable to servers or to applications from other vendors.

<47> Section 2.7.1.1.1.1.1.1.2: The contents of the **ObjectRef** string apply only to the Microsoft Object Model, AMO.

<48> Section 2.7.1.1.1.1.1.1.2: The contents of the **ObjectRef** string applies only to the Microsoft Object Model, AMO.

<49> Section 2.7.1.1.1.1.1.1.2: In Microsoft implementations, the value -1 indicates that no value is assigned to the **BulkInsertFirstRow** attribute.

<50> Section 2.7.1.1.1.1.1.1.2: In Microsoft implementations, if the value of the **LastRow** attribute is -1, the value is not set.

<51> Section 2.7.1.1.1.1.1.1.2: In Microsoft implementations, the value 0 for the **Timeout** attribute indicates that SQL Server does not time out.

<52> Section 2.7.1.1.1.1.1.1.2: In Microsoft implementations, when the **MaxInsertCommitSize** attribute value is set to 0, all rows are inserted in a single batch.

<53> Section 2.7.1.1.1.1.1.1.2: n Microsoft implementations, the value of 0 for the **CommandTimeout** attribute indicates that there is no timeout.

<54> Section 2.7.1.1.1.1.1.1.2: For information about how Integration Services converts external types to strings, see [MSDN-ADONETSRC].

<55> Section 2.7.1.1.1.1.1.1.2: In Microsoft implementations, the value of 0 for the **CommandTimeout** attribute indicates that there is no timeout.

<56> Section 2.7.1.1.1.1.1.1.2: In Microsoft implementations, the value of 0 for the **CommandTimeout** attribute indicates that there is no timeout.

<57> Section 2.7.1.1.1.1.1.2.1: The Integration Services UI generates a unique value for the **refId** attribute value at the time of object creation.

<58> Section 2.7.1.1.1.1.3.1.3: In Microsoft implementations, a value of -1 for the **DestinationColumn** property indicates that the column is not mapped to an output column.

<59> Section 2.7.1.1.1.1.4.1.1: In Microsoft implementations, a value of -1 for the **SourceColumn** attribute indicates that the column is not used for pivoting.

<60> Section 2.7.1.1.1.1.5: In Microsoft implementations, the list of components for the **PipelineComponentClassIDEnum** value of "{874F7595-FB5F-40FF-96AF-FBFF8250E3EF}" can be distinguished by the **UserComponentTypeName** attribute value on the component element.

<61> Section 2.7.1.1.1.1.6.3: The Integration Services user interface generates a unique value for the **refId** attribute value at the time of object creation.

<62> Section 2.7.1.1.1.1.6.4: The Integration Services user interface generates a unique value for the **refId** attribute value at the time of object creation.

<63> Section 2.7.1.1.1.1.6.5: This value is assigned by the Integration Services engine when a column is created and is used internally by the Integration Services engine to track the relationships between input columns and output columns that are downstream from this column and that reference this column.

<64> Section 2.7.1.1.2.1: The Integration Services user interface generates a unique value for the **refId** attribute value at the time of object creation.

<65> Section 2.7.1.5.1.2: In Microsoft implementations, the value 0 for **BatchSize** indicates default behavior, which is that all rows in the data file are processed as one transaction.

<66> Section 2.7.1.5.1.2: In Microsoft implementations, a value of 0 for the **LastRow** attribute indicates default behavior, which is to process to the end of the file.

<67> Section 2.7.1.5.1.2.1: In SQL Server, for the DTSBulkInsert_DataFileType_Native enumeration value, the data file to load is created by bulk copying SQL Server data by using the bcp utility. For more information about the bcp utility, see [MSDN-BCPU].

<68> Section 2.7.1.5.1.2.1: In SQL Server, for the DTSBulkInsert_DataFileType_WideNative enumeration value, the data file to load is created by bulk copying SQL Server data by using the bcp utility.

<69> Section 2.7.1.5.1.2.2: The Bulk Insert Task works only in conjunction with SQL Server as the server. When the **CodePageType** attribute is set to the value "ACP", the code page is converted from the ANSI code page to the SQL Server code page.

<70> Section 2.7.1.5.1.2.2: The Bulk Insert Task works only in conjunction with SQL Server as the server. When the **CodePageType** attribute is set to the value "OEM", the code page is converted to the SQL Server code page.

<71> Section 2.7.1.6: In Microsoft implementations, this string is in the Data Mining Extensions (DMX) [MSDN-DMXR] language.

<72> Section 2.7.1.7.1.1.1: The contents of the **DataSourcesType** type are dynamically generated by Integration Services and contain a reference to each **ConnectionManager** in the Integration Services package file. If any contents are present in this element in the package file, the content is ignored and dynamically regenerated as just stated.

<73> Section 2.7.1.7.1.1.2: For Integration Services, the **ProfileMode** element has to be set to the value "Exact". Fast mode is not supported.

<74> Section 2.7.1.7.1.1.2.2.2: By default, in Microsoft implementations, if the **FrequentValueThreshold** value is not specified, Integration Services sets the value to 0.001.

<75> Section 2.7.1.7.1.1.2.2.2: In Microsoft implementations, if the **MaxNumberOfPatterns** element is not specified, the default value is 10.

<76> Section 2.7.1.7.1.1.2.2.2: In Microsoft implementations, if the **PercentageDataCoverageDesired** element is not specified, the default value is 95.

<77> Section 2.7.1.7.1.1.2.2.2: In Microsoft implementations, if the **CaseSensitive** element is not specified, the default value is FALSE.

<78> Section 2.7.1.7.1.1.2.2.2: By default, in Microsoft implementations, the list of **Delimiter** elements contains the following characters:

▪ Space

- Horizontal tab (\t)

- New line (\n)

- Carriage return (\r)

Additional **Delimiter** elements can be specified, but the default delimiters cannot be removed.

<79> Section 2.7.1.7.1.1.2.2.2: By default, in Microsoft implementations, the list of symbols contains the following characters: , . ; : - " ' ` ~ = & / \ @ ! ? ( ) < > [ ] { } | # ^ %

<80> Section 2.7.1.7.1.1.2.2.2: By default, in Microsoft implementations, if the **KeyStrengthThreshold** element value is not specified, the value of 0.95 is used.

<81> Section 2.7.1.7.1.1.2.2.2: Integration Services supports the value FALSE only for the **VerifyOutputInFastMode** element.

<82> Section 2.7.1.7.1.1.2.2.2: In Microsoft implementations, the default value for the **MaxNumberOfViolations** element is 100.

<83> Section 2.7.1.7.1.1.2.2.2: In Microsoft implementations, the default value of 0.95 is used for the value of the **FDStrengthThreshold** element if the value of the **FDStrengthThreshold** element is not specified.

<84> Section 2.7.1.7.1.1.2.2.2: Integration Services supports the value FALSE only for the **VerifyOutputInFastMode** element.

<85> Section 2.7.1.7.1.1.2.2.2: In Microsoft implementations, the default value for the **MaxNumberOfViolations** element is 100.

<86> Section 2.7.1.7.1.1.2.2.2: In Microsoft implementations, the **SupersetDataSourceID** attribute has to be equal to the **SubsetDataSourceID** attribute.

<87> Section 2.7.1.7.1.1.2.2.2: By default, in Microsoft implementations, if the value of the **InclusionStrengthThreshold** element is not specified, the value 0.95 is used.

<88> Section 2.7.1.7.1.1.2.2.2: By default, in Microsoft implementations, if the value of the **SupersetColumnsKeyThreshold** element is not specified, the value 0.95 is used.

<89> Section 2.7.1.7.1.1.2.2.2: Integration Services supports the value FALSE only for the **VerifyOutputInFastMode** element.

<90> Section 2.7.1.7.1.1.2.2.2: In Microsoft implementations, the default value for the **MaxNumberOfViolations** element is 100.

<91> Section 2.7.1.7.1.1.3: In Microsoft implementations, the contents of the **DataProfileOutputType** complex type are not used in the package file. If the **DataProfileOutputType** complex type is present, any content is ignored.

<92> Section 2.7.1.7.1.1.3: In Microsoft implementations, the **Profiles** element is enforced to be empty. Any content that is present is ignored during package file processing.

<93> Section 2.7.1.8.1.1.1: In Microsoft implementations, every time the package is saved, a new GUID value is generated for the **PackageVersionGUID** attribute.

<94> Section 2.7.1.8.1.1.1: In Microsoft implementations, the **SQLServer** attribute value is the name of an instance of SQL Server.

<95> Section 2.7.1.8.1.1.1: In Microsoft implementations, the database username is a specific SQL Server username. The **SQLUsername** attribute value is used only if the **WindowsAuthentication** attribute has a value of "False".

<96> Section 2.7.1.8.1.1.1.1: In Microsoft implementations, the "SQLServer" enumeration value specifies that the DTS 2000 Package file is stored on a SQL Server Database server.

<97> Section 2.7.1.11.1.1.1.8: In Microsoft implementations, if a value for the **CodePage** attribute is not specified, the default value that is set in the Windows operating system is obtained from the operating system and used.

<98> Section 2.7.1.14.1.1.1: The **DTSMessageLineageID** attribute value is not used by the Integration Services system. Any value given for this attribute is ignored by Integration Services.

<99> Section 2.7.1.14.1.1.1: In Microsoft implementations, the **Use2000Format** attribute specifies whether the message is formatted with the Microsoft SQL Server 2000 format:

▪ "True" specifies that the message is formatted with the SQL Server 2000 format.

▪ "False" specifies that the message is not formatted with the SQL Server 2000 format.

<100> Section 2.7.1.15: In Microsoft implementations, the **VSTAMajorVersion** integer value specifies the Visual Studio Tools for Applications (VSTA) major version number.

<101> Section 2.7.1.15: In Microsoft implementations, the **VSTAMinorVersion** integer value specifies the VSTA minor version number.

<102> Section 2.7.1.15.3: This string value is used by Visual Studio for supporting the debugging environment and is not set or interpreted by Integration Services.

<103> Section 2.7.1.22: Database logins are the logins authorized to the SQL Server instance.

<104> Section 2.8.5.1: The Integration Services user interface generates a unique value for the **refId** attribute value at the time of object creation.

<105> Section 2.8.5.1.1: The content of the **EventID** attribute value is ignored by the Integration Services system.

<106> Section 2.8.6.1: The Integration Services user interface generates a unique value for the **refId** attribute value at the time of object creation.

<107> Section 2.9.1: The content of the **Description** string is not processed by the Integration Services system.

<108> Section 2.9.1: The Integration Services user interface generates a random unique value for the **DTSID** attribute value.

<109> Section 2.9.1: The Integration Services user interface generates a value for the **CreationName** attribute value at the time of object creation.

<110> Section 2.9.1: The Integration Services user interface generates a value for the **ObjectName** attribute value at the time of object creation.

<111> Section 2.9.2: In Microsoft implementations, if a value for the LCID attribute is not specified, the default value that is set in the Windows operating system is obtained from the system and used.

<112> Section 2.9.10: The **InnerObjectObjectDataType** type is empty for all standard components that ship with the Integration Services product.

# 7 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as Major, Minor, or None.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements.
- A document revision that captures changes to protocol functionality.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **None** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the relevant technical content is identical to the last released version.

The changes made to this document are listed in the following table. For more information, please contact dochelp@microsoft.com.

| Section | Description | Revision class |
|---|---|---|
| 1.6 Versioning and Localization | Added SQL Server 2022 to the product behavior note. | Minor |

# 8   Index

**A**

**I**

**L**

**N**

**O**

**P**

**R**

**S**

**T**

**U**

**V**

**W**