[MS-DPIS]:
Integration Services Data Portability Overview

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **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 may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft Open Specification Promise or the Community Promise. If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.
- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events 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 specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications do 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 are intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

Revision Summary

Date	Revision History	Revision Class	Comments
02/09/2011	0.1	New	Released new document.
07/07/2011	1.0	Major	Significantly changed the technical content.
11/03/2011	1.0	No change	No changes to the meaning, language, or formatting of the technical content.

Contents

1 Introduction

The Microsoft SQL Server Integration Services system includes a repository for data integration artifacts, including Integration Services packages (as documented in [MS-DTSX] and [MS-DTSX2]), Data Transformation Services (DTS) packages, and Integration Services project deployment files (as documented in [MS-DTS] and [MS-ISPAC]). This repository includes a set of Microsoft® SQL Server® tables, views, and stored procedures, depending on the format that is used, as described in the following table.

Artifact	Customary file system extension	Documented in	SQL Server object	SQL Server database
Data Transformation Services Package XML package	.dtsx	[MS-DTSX] and [MS-DTSX2]	**sysssispackages** (table)	**msdb**
Data Transformation Services package	.dts	[MS-DTS]	**sysdtspackages** (table)	**msdb**
Integration Services project deployment file	.ispac	[MS-ISPAC]	**catalog.projects** (view) **catalog.get_project** (stored procedure)	**SSISDB**

Data integration artifacts are set or retrieved in this database by using SQL statements that are transmitted over the SQL Server Tabular Data Stream (TDS) protocol [MS-TDS]. Or, the artifacts are opened or saved in the file system, where they exist as files that are named with an extension that is based on the customary file system extension, as described in the preceding table.

1.1 Glossary

1.2 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. Please check the archive site, http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624, as an additional source.

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

[MS-DTSX2] Microsoft Corporation, "Data Transformation Services Package XML Version 2 File Format Specification".

[MS-ISPAC] Microsoft Corporation, "Integration Services Project Deployment File Format Structure Specification".

[MS-TDS] Microsoft Corporation,"Tabular Data Stream Protocol Specification".

2 Data Portability Scenarios

2.1 Third-Party Integration Tool or Platform Consuming Integration Services Packages from MSDB Repository

2.1.1 Data Description

The DTSX documentation [MS-DTSX] and [MS-DTSX2] contain the definition of a package, which includes information about configured connection managers, data sources, destinations, and transformations that should be applied to data, as well as the ordering of various tasks that are involved in an extraction, transformation, and loading (ETL) package. This data is represented in the DTSX document as XML, as documented in [MS-DTSX] and [MS-DTSX2].

This DTSX data is used to effect a set of data movements and transformations, typically from one or more sources to one or more destinations as configured in the package. This DTSX data is created by using the Business Intelligence Design Studio that is included with Microsoft® SQL Server® or by using the **Microsoft.SqlServer.Dts** object model.

DTSX data may be stored as a file on the file system or in the **msdb** repository as a row in the **sysssispackages** table.

2.1.2 Format and Protocol Summary

The following table provides a comprehensive list of the formats and protocols used in this scenario.

Format or protocol name	Description	Short name
TDS	This protocol is used to communicate with Microsoft® SQL Server® to execute SQL statements and retrieve results.	[MS-TDS]

2.1.3 Data Portability Methodology

For this scenario, the documents that contain the DTSX data are extracted from the **msdb** repository one by one and stored in a file on the file system. The method of extracting the DTSX data from the **msdb** repository for use in a third-party integration tool or platform in this scenario is to use the TDS protocol that is provided by the Microsoft® SQL Server® instance.

To extract the data, follow these steps:

1. Create a folder on the client machine for storing the retrieved DTSX documents.
2. Connect and authenticate to the desired server by using an ODBC, OLEDB, or ADO.NET provider or any other TDS implementation.
3. Issue the following SQL statement:

   ```
   SELECT name, packagedata from msdb..sysssispackages
   ```

4. Store the data. To do this, follow these steps for each row that is returned in step 3:
   1. Create a new file in the folder that was created in step 1.
   2. Save the contents of the **packagedata** field to the new file.

5. Use the DTSX documentation [MS-DTSX] or [MS-DTSX2] to interpret the DTSX data that was retrieved in the previous step for use in the third-party integration tool or platform.

2.1.3.1 Preconditions

Ensure that the Microsoft® SQL Server® service is started on the server. Grant the appropriate permissions to the user to access the **sysssispackages** table.

2.1.3.2 Versioning

None.

2.1.3.3 Error Handling

None.

2.1.3.4 Coherency Requirements

This data portability scenario has no special coherency requirements.

2.1.3.5 Additional Considerations

There are no additional considerations.

2.2 Third-Party Integration Platform or Tool Consuming Integration Services Packages in the File System

2.2.1 Data Description

The DTSX documentation [MS-DTSX] and [MS-DTSX2] contain the definition of a package, which includes information about configured connection managers, data sources, destinations, and transformations that should be applied to data, and the ordering of various tasks that are involved in an ETL package. This data is represented in the DTSX document as XML, as documented in [MS-DTSX] and [MS-DTSX2].

This DTSX data is used to effect a set of data movements and transformations, typically from one or more sources to one or more destinations as configured in the package. This data is created by using the Business Intelligence Design Studio that is included with Microsoft® SQL Server® or by using the **Microsoft.SqlServer.Dts** object model.

2.2.2 Format and Protocol Summary

No formats or protocols are used in this scenario.

2.2.3 Data Portability Methodology

In this scenario, the DTSX data is stored in the file system as packages (*.dtsx files). These files may be stored at any location on a system. Use the DTSX documentation [MS-DTSX] or [MS-DTSX2] to interpret the DTSX data in these files.

2.2.3.1 Preconditions

None.

2.2.3.2 Versioning

None.

2.2.3.3 Error Handling

None.

2.2.3.4 Coherency Requirements

This data portability scenario has no special coherency requirements.

2.2.3.5 Additional Considerations

There are no additional considerations.

2.3 Third-Party Integration Tool or Platform Consuming DTS Packages from MSDB Repository

2.3.1 Data Description

The DTS [MS-DTS] document contains the definition of a DTS package, which includes information about the configuration and order of tasks and data pumps that are applied to data in a DTS package. This data is represented in the DTS document as a binary file, as documented in [MS-DTS].

This DTS data is used to effect a set of data movements and transformations, typically from one or more sources to one or more destinations as configured in the package. This data is created by using the Business Intelligence Design Studio that is included with Microsoft® SQL Server® or by using SQL Server Enterprise Manager in Microsoft® SQL Server® 2000.

DTS data may be stored as a file on the file system or in the repository as a row in the **sysdtspackages** table in the **msdb** repository.

2.3.2 Format and Protocol Summary

The following table provides a comprehensive list of the formats and protocols used in this scenario.

Format or protocol name	Description	Short name
TDS	This protocol is used to communicate with Microsoft® SQL Server® to execute SQL statements and retrieve results.	[MS-TDS]

2.3.3 Data Portability Methodology

For this scenario, the documents that contain the DTS data are extracted from the **msdb** repository one by one and stored in a file on the file system. The method of extracting the DTS data from the **msdb** repository for use in a third-party integration tool or platform in this scenario is to use the TDS protocol that is provided by the Microsoft® SQL Server® instance.

To extract the data, follow these steps:

1. Create a folder on the client machine for storing the retrieved DTS documents.

2. Connect and authenticate to the desired server by using an ODBC, OLEDB, or ADO.NET provider or any other TDS implementation.

3. Issue the following SQL statement:

   "SELECT name, packagedata from msdb..sysdtspackages"

4. Store the data. To do this, follow these steps for each returned row in step 3:

   1. Create a new file in the folder that was created in step 1.

   2. Save the contents of the **packagedata** field to the new file.

5. Use the DTS documentation [MS-DTS] to interpret the DTS data that was retrieved in the previous step for use in the third-party integration tool or platform.

2.3.3.1 Preconditions

Ensure that the Microsoft® SQL Server® service is started on the server. Grant the appropriate permissions to the user to access the **sysdtspackages** table.

2.3.3.2 Versioning

Each DTS document contains multiple versions of the DTS package, as documented in [MS-DTS].

2.3.3.3 Error Handling

None.

2.3.3.4 Coherency Requirements

This data portability scenario has no special coherency requirements.

2.3.3.5 Additional Considerations

There are no additional considerations.

2.4 Third-Party Integration Platform or Tool Consuming DTS Packages in the File System

2.4.1 Data Description

The DTS [MS-DTS] document contains the definition of a DTS package, which includes information about the configuration and order of tasks and data pumps that are applied to data in a DTS package. This data is represented in the DTS document as a binary file, as documented in [MS-DTS].

This DTS data is used to effect a set of data movements and transformations, typically from one or more sources to one or more destinations as configured in the package. This data is created by using the Business Intelligence Design Studio that is included with Microsoft® SQL Server® or by using SQL Server Enterprise Manager in Microsoft® SQL Server® 2000.

2.4.2 Format and Protocol Summary

No formats or protocols are used in this scenario.

2.4.3 Data Portability Methodology

In this scenario, the DTS data is stored in the file system as packages (*.dts files). These files may be stored at any location on a system. Use the DTS documentation [MS-DTS] to interpret the DTS data in these files.

2.4.3.1 Preconditions

None.

2.4.3.2 Versioning

Each DTS document contains multiple versions of the DTS package as documented in [MS-DTS].

2.4.3.3 Error Handling

None.

2.4.3.4 Coherency Requirements

This data portability scenario has no special coherency requirements.

2.4.3.5 Additional Considerations

There are no additional considerations.

2.5 Third-Party Integration Tool or Platform Consuming Integration Services Project Deployment Files from SSISDB Repository

2.5.1 Data Description

An ISPAC document contains the definition of an Integration Services project deployment file, which includes the packaged metadata of a data integration project. This data is represented in the ISPAC document, as documented in [MS-ISPAC].

This ISPAC data is used to package a set of interrelated metadata that describes one or more data integration processes. This data is created by using the Business Intelligence Design Studio that is included with Microsoft® SQL Server® or by using the **Microsoft.SqlServer.Dts** object model.

ISPAC data may be stored as a file on the file system or in the repository as a row that is accessible through the catalog.projects view in the **SSISDB** database.

2.5.2 Format and Protocol Summary

The following table provides a comprehensive list of the formats and protocols used in this scenario.

Format or protocol name	Description	Short name
TDS	This protocol is used to communicate with Microsoft® SQL Server® to execute SQL statements and retrieve results.	[MS-TDS]

2.5.3 Data Portability Methodology

For this scenario, the documents containing the ISPAC data are extracted from the **SSISDB** database one by one and stored in a file on the file system. The method of extracting the ISPAC data from the **SSISDB** database for use in a third-party integration tool or platform in this scenario is to use the TDS protocol that is provided by the Microsoft® SQL Server® instance.

To extract the data, follow these steps:

1. Create a folder on the client machine for storing the retrieved ISPAC documents.
2. Connect and authenticate to the desired server by using an ODBC, OLEDB, or ADO.NET provider or any other TDS implementation.
3. Issue the following SQL statement:

```
SELECT P.name as project_name, F.name as folder_name from
ssisdb.catalog.projects P INNER JOIN ssisdb.catalog.folders F on
F.folder_id=P.folder_id
```

4. Store the data. To do this, follow these steps for each returned row in step 3:
    1. Create a new file in the folder that was created in step 1.
    2. Invoke the **catalog.get_project** stored function, passing the value of the **project_name** and **folder_name** fields into the *@project_name* and *@folder_name* parameters, respectively.
    3. Save the contents of the return value from **catalog.get_project** to the new file.
5. Use the ISPAC documentation [MS-ISPAC] to interpret the ISPAC data that was retrieved in the previous step for use in the third-party integration tool or platform.

2.5.3.1 Preconditions

Ensure that the Microsoft® SQL Server® service is started on the server. Grant the appropriate permissions to the user to access the **SSISDB** catalog views.

2.5.3.2 Versioning

None.

2.5.3.3 Error Handling

None.

2.5.3.4 Coherency Requirements

This data portability scenario has no special coherency requirements.

2.5.3.5 Additional Considerations

There are no additional considerations.

2.6 Third-Party Integration Platform or Tool Consuming Integration Services Project Deployment Files Packages in the File System

2.6.1 Data Description

An ISPAC document contains the definition of an Integration Services project deployment file, which includes the packaged metadata of a data integration project. This data is represented in the ISPAC document, as documented in [MS-ISPAC].

This ISPAC data is used to package a set of interrelated metadata that describes one or more data integration processes. This data is created by using the Business Intelligence Design Studio that is included with Microsoft® SQL Server® or by using the **Microsoft.SqlServer.Dts** object model.

ISPAC data may be stored as a file on the file system or in the repository as a row that is accessible through the **catalog.projects** view in the **SSISDB** database.

2.6.2 Format and Protocol Summary

No formats or protocols are used in this scenario.

2.6.3 Data Portability Methodology

In this scenario, the ISPAC data is stored in the file system as packages (*.ispac files). These files may be stored at any location on a system. Use the ISPAC documentation [MS-ISPAC] to interpret the ISPAC data in these files.

2.6.3.1 Preconditions

None.

2.6.3.2 Versioning

None.

2.6.3.3 Error Handling

None.

2.6.3.4 Coherency Requirements

This data portability scenario has no special coherency requirements.

2.6.3.5 Additional Considerations

There are no additional considerations.

3 Change Tracking Page

No table of changes is available. The document is either new or has had no changes since its last release.

4 Index