[MS-DPBACPAC]: Data-Tier Application Schema and Data Data Portability Overview

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
07/07/2011	0.1	New	Released new document.
11/03/2011	0.1	No change	No changes to the meaning, language, or formatting of the technical content.
01/19/2012	0.1	No change	No changes to the meaning, language, or formatting of the technical content.
02/23/2012	0.1	No change	No changes to the meaning, language, or formatting of the technical content.
03/27/2012	0.1	No change	No changes to the meaning, language, or formatting of the technical content.
05/24/2012	0.1	No change	No changes to the meaning, language, or formatting of the technical content.
06/29/2012	0.1	No change	No changes to the meaning, language, or formatting of the technical content.
07/16/2012	0.1	No change	No changes to the meaning, language, or formatting of the technical content.
10/08/2012	1.0	Major	Significantly changed the technical content.

Contents

1 Introduction 4
1.1 Glossary 4
1.2 References... 5

2 Data Portability Scenarios 6
2.1 Retrieve Data and Schema from .bacpac File.. 6
2.1.1 Data Description 6
2.1.2 Format and Protocol Summary 6
2.1.3 Data Portability Methodology 7
2.1.3.1 Preconditions 7
2.1.3.2 Versioning 7
2.1.3.3 Error Handling 7
2.1.3.4 Coherency Requirements 7
2.1.3.5 Additional Considerations... 7

3 Change Tracking.. 8

4 Index 11

1 Introduction

A data-tier application component (DAC) is a self-contained unit of database deployment and management that enables data-tier developers and database administrators (DBAs) to package Microsoft SQL Server objects, including **database** and instance objects, into a single entity called a DAC package (.dacpac or .bacpac file), as specified in [MSDN-DACOVERVIEW]. A .dacpac file is a package of XML parts that represent:

- Metadata of the data-tier application and SQL Server object the **schema** of the SQL Server database that the package represents; and
- Data from the source representation (project, database, scripts).

The BACPAC file format extends the DACPAC file format to include all table data, in addition to schema data.

Both .dacpac and .bacpac files are Open Packaging Conventions (OPC) packages that comply with the OPC specification [ECMA-376-2/2]. A .dacpac file can optionally also contain data from a set of user tables defined in the schema. Procedures similar to those defined in this document also apply to .dacpac packages that contain user table data. For more information, see [MSDN-DACDeploy].

This document provides an overview of the data portability scenario to retrieve data and schema from a .bacpac file. In this scenario, a vendor must use the Microsoft Data-Tier Application Framework (DACFx) API [MSDN-DACAPI-3] or SQLPackage.exe to consume the .bacpac file.

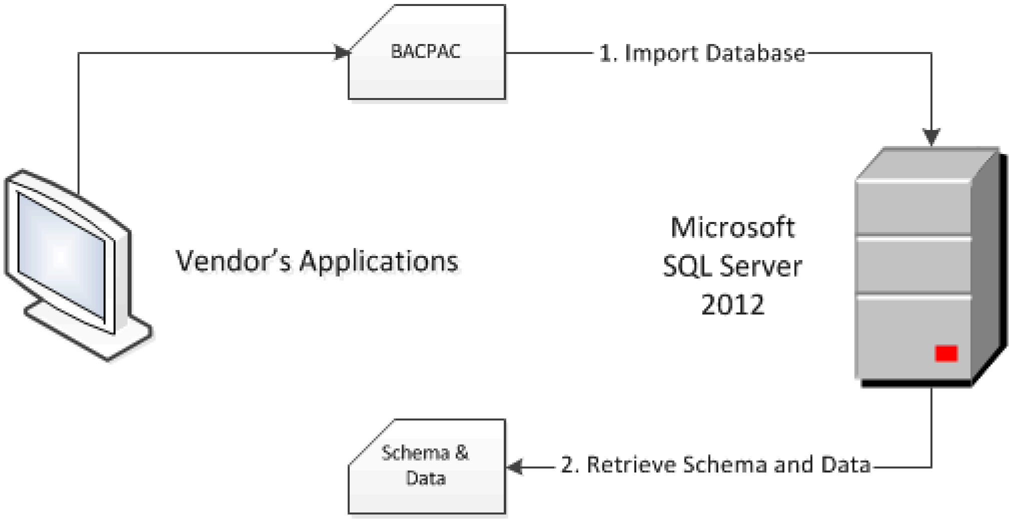

Figure 1: Conceptual overview of .bacpac data portability

A vendor can implement an application by using the DACFx API to create a .bacpac file and import that file into SQL Server or Microsoft Windows Azure SQL Database. Then, the vendor can retrieve the schema and data as described in section 2.1.

1.1 Glossary

The following terms are defined in [MS-GLOS]:

database
schema

1.2 References

[ECMA-376-2/2] ECMA, "Office Open XML File Formats – Part 2: Open Packaging Conventions", 2nd edition, ECMA-376-2, December 2008, http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-376,%20Second%20Edition,%20Part%202%20-%20Open%20Packaging%20Conventions.zip

[MS-GLOS] Microsoft Corporation, "Windows Protocols Master Glossary".

[MSDN-BCPU] Microsoft Corporation, "bcp Utility", http://msdn.microsoft.com/en-us/library/ms162802(SQL.105).aspx

[MSDN-CREATEFF] Microsoft Corporation, "Create a Format File (SQL Server)", http://msdn.microsoft.com/en-us/library/ms191516.aspx

[MSDN-DACAPI-3] Microsoft Corporation, "Microsoft.SqlServer.Dac Namespace", http://msdn.microsoft.com/en-us/library/hh753459.aspx

[MSDN-DACDeploy] Microsoft Corporation, "Deploy a Data-tier Application", http://msdn.microsoft.com/en-us/library/ee210569.aspx

[MSDN-DACOVERVIEW] Microsoft Corporation, "Data-tier Applications", http://msdn.microsoft.com/en-us/library/ee210546.aspx

[MSDN-DacSvcsExc] Microsoft Corporation, "DacServicesException Class", http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.dac.dacservicesexception.aspx

[MSDN-DacSvcsImport] Microsoft Corporation, "DacServices.ImportBacpac Method", http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.dac.dacservices.importbacpac.aspx

[MSDN-FFIED] Microsoft Corporation, "Format Files for Importing or Exporting Data (SQL Server)", http://msdn.microsoft.com/en-us/library/ms190393.aspx

[MSDN-SIOPN] Microsoft Corporation, "System.IO.Packaging Namespace", http://msdn.microsoft.com/en-us/library/system.io.packaging.aspx

2 Data Portability Scenarios

2.1 Retrieve Data and Schema from .bacpac File

The retrieve data and schema scenario describes importing a .bacpac file that contains database schema and table data to a Microsoft SQL Server database or Microsoft Windows Azure SQL Database, as shown in the following figure.

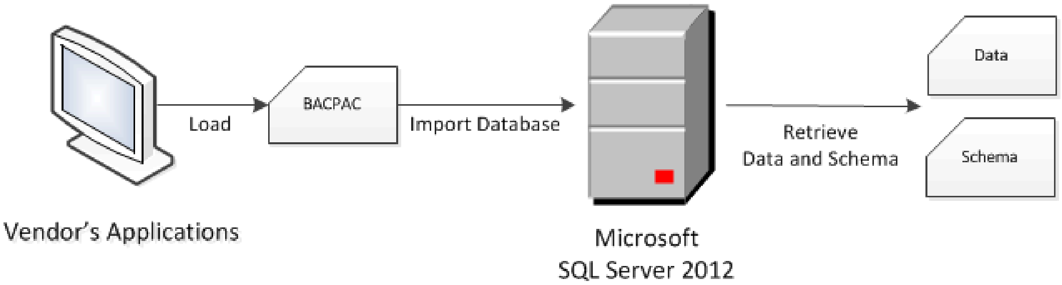

Figure 2: Retrieve data and schema from .bacpac file

2.1.1 Data Description

Customer data

Customer data is a schema representation of a Microsoft SQL Server database or Microsoft Windows Azure SQL Database and user data contained within the tables that are included in the package. Customer data is serialized into the .bacpac by using the Bulk Copy Program (BCP) file format [MSDN-FFIED].

Intended user

The intended user is a customer or vendor who can import SQL Server object schema and table data from a .bacpac file to a SQL Server or Windows Azure SQL Database and then retrieve the data and schema.

2.1.2 Format and Protocol Summary

The following table provides a comprehensive list of the formats and protocols used in a retrieve data and schema data portability scenario.

Format name	Description	Reference
Data-Tier Application File (BACPAC) Format	The BACPAC file format serves as the packaging format for an exported database that contains database schema and user data.	None.
Microsoft.SqlServer.Dac Namespace	The **Microsoft.SqlServer.Dac** namespace contains classes that represent DAC Framework objects.	[MSDN-DACAPI-3]
System.IO.Packaging Namespace	The **System.IO.Packaging** namespace of the .NET Framework provides classes that support storage of multiple data objects in a single container.	[MSDN-SIOPN]
Bulk Copy File Format	The Bulk Copy File Format is used for the data streams contained within the package.	[MSDN-FFIED]

2.1.3 Data Portability Methodology

The data portability methodology describes the packaging and deployment steps to take when using the DACFx API [MSDN-DACAPI-3].

Import a data-tier application

To import a data-tier application, load the .bacpac file, and then import it to a Microsoft SQL Server database or Microsoft Windows Azure SQL Database. For more information about the **DacServices.ImportBacpac** method, see [MSDN-DacSvcsImport].

Retrieve data and schema

The vendor can then use a variety of methods to retrieve the database schema and user data from the database into the file format desired. This document describes one possible method of doing so, which consists of using the Bulk Copy Program (BCP) Utility [MSDN-BCPU].

1. First, use bcp.exe to create a format file to record the data format as described in [MSDN-BCPU] and [MSDN-CREATEFF]. The bcp utility supports XML and other file formats.

2. Then, use bcp.exe to extract table data as per the format file.

The BCP file format is described in further detail in [MSDN-FFIED].

2.1.3.1 Preconditions

The Microsoft SQL Server user must be a member of the **dbcreator** fixed server role on the SQL Server instance to import the .bacpac file. The Microsoft Windows Azure SQL Database user must be a member of the **dbmanager** fixed logical server role.

2.1.3.2 Versioning

This version of the retrieve data and schema scenario is applicable to Microsoft SQL Server 2005 Service Pack 2 (SP2), Microsoft SQL Server 2008 Service Pack 1 (SP1), Microsoft SQL Server 2008 R2 Service Pack 1 (SP1), Microsoft SQL Server 2012, and Microsoft Windows Azure SQL Database.

2.1.3.3 Error Handling

The data-tier application error and exception class is described in [MSDN-DacSvcsExc].

2.1.3.4 Coherency Requirements

When importing a .bacpac file to a database, the database must either not exist or be a new, empty database. Otherwise, the import operation will fail.

2.1.3.5 Additional Considerations

There are no additional considerations.

3 Change Tracking

This section identifies changes that were made to the [MS-DPBACPAC] protocol document between the July 2012 and October 2012 releases. Changes are classified as New, Major, Minor, Editorial, or No change.

The revision class **New** means that a new document is being released.

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 or functionality.
- An extensive rewrite, addition, or deletion of major portions of content.
- The removal of a document from the documentation set.
- Changes made for template compliance.

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 **Editorial** means that the language and formatting in the technical content was changed. Editorial changes apply to grammatical, formatting, and style issues.

The revision class **No change** means that no new technical or language changes were introduced. The technical content of the document is identical to the last released version, but minor editorial and formatting changes, as well as updates to the header and footer information, and to the revision summary, may have been made.

Major and minor changes can be described further using the following change types:

- New content added.
- Content updated.
- Content removed.
- New product behavior note added.
- Product behavior note updated.
- Product behavior note removed.
- New protocol syntax added.
- Protocol syntax updated.
- Protocol syntax removed.
- New content added due to protocol revision.
- Content updated due to protocol revision.
- Content removed due to protocol revision.
- New protocol syntax added due to protocol revision.

- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.
- New content added for template compliance.
- Content updated for template compliance.
- Content removed for template compliance.
- Obsolete document removed.

Editorial changes are always classified with the change type **Editorially updated.**

Some important terms used in the change type descriptions are defined as follows:

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.
- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

The changes made to this document are listed in the following table. For more information, please contact protocol@microsoft.com.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
1 Introduction	Revised content to describe how the user can retrieve assets from the packages without needing to interpret the BACPAC or DACPAC format.	N	Content updated.
1.1 Glossary	Removed the term "JavaScript Object Notation (JSON)" from the list of document-specific terms.	N	Content updated.
1.2 References	Added references [ECMA-376-2/2], [MS-GLOS], [MSDN-BCPU], [MSDN-CREATEFF], [MSDN-DACAPI-3], [MSDN-DACDeploy], [MSDN-DACOVERVIEW], [MSDN-DacSvcsExc], [MSDN-DacSvcsImport], [MSDN-FFIED], and [MSDN-SIOPN], and removed references [MS-BACPAC], [MS-DACPAC], [MSDN-DACAPI-2], [MSDN-DACSTOR], [MSDN-DacStore.Export], [MSDN-DacStore.Import], [MSDN-DACSUPOB-2], [MSDN-DBSTATE-2], [MSDN-JSONSer], [MSDN-PACKGET], [MSDN-PACKNAME], [MSDN-PACKOP], [MSDN-PACKPARTCON], [MSDN-TDAC], [MSDN-UNDERDAC-2], and [RFC4627].	N	Content updated.
2.1 Retrieve Data and Schema from .bacpac File	Changed section title from "Export Data" to "Retrieve Data and Schema from .bacpac File" and revised content of section and its subsections to describe how the user can retrieve assets from the packages without needing to interpret the BACPAC or DACPAC format.	Y	Content updated.
2.1.1 Data Description	Revised content to describe how the user can retrieve assets from the packages without needing to interpret the BACPAC or DACPAC format.	Y	Content updated.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
2.1.2 Format and Protocol Summary	Revised content to describe how the user can retrieve assets from the packages without needing to interpret the BACPAC or DACPAC format.	Y	Content updated.
2.1.3 Data Portability Methodology	Revised content to describe how the user can retrieve assets from the packages without needing to interpret the BACPAC or DACPAC format.	Y	Content updated.
2.1.3.1 Preconditions	Revised content to describe how the user can retrieve assets from the packages without needing to interpret the BACPAC or DACPAC format.	Y	Content updated.
2.1.3.2 Versioning	Revised content to describe how the user can retrieve assets from the packages without needing to interpret the BACPAC or DACPAC format.	Y	Content updated.
2.1.3.3 Error Handling	Revised content to describe how the user can retrieve assets from the packages without needing to interpret the BACPAC or DACPAC format.	Y	Content updated.
2.1.3.4 Coherency Requirements	Revised content to describe how the user can retrieve assets from the packages without needing to interpret the BACPAC or DACPAC format.	Y	Content updated.
2.2 Import Data	Removed section 2.2 and all its subsections.	Y	Content removed.

4 Index

C

Change tracking 8

G

glossary 4

R

references 5
Retrieve data and schema scenario 6

T

Tracking changes 8