**Data-tier Applications**

**SQL Server 2012**

[Other Versions](javascript:;)

Description: http://i.technet.microsoft.com/Areas/Epx/Content/Images/ImageSprite.png

* [SQL Server 2008 R2](http://technet.microsoft.com/en-us/library/ee210546(d=printer,v=sql.105).aspx)
* [SQL Server 2008](http://technet.microsoft.com/en-us/library/ee210546(d=printer,v=sql.100).aspx)

A data-tier application (DAC) is a logical database management entity that defines all of the SQL Server objects - like tables, views, and instance objects, including logins – associated with a user’s database. A DAC is a self-contained unit of SQL Server database deployment that enables data-tier developers and database administrators to package SQL Server objects into a portable artifact called a DAC package, also known as a DACPAC.

A BACPAC is a related artifact that encapsulates the database schema as well as the data stored in the database.

[Benefits of Data-tier Applications](javascript:void(0))

The lifecycle of most database applications involves developers and DBAs sharing and exchanging scripts and ad hoc integration notes for application update and maintenance activities. While this is acceptable for a small number of databases, it quickly becomes unscalable once databases grow in number, size, and complexity.

A DAC is a database lifecycle management and productivity tool that enables declarative database development to simplify deployment and management. A developer can author a database in SQL Server Data Tool database project and then build the database into a DACPAC for handoff to a DBA. The DBA can deploy the DAC using SQL Server Management Studio to a test or production instance of SQL Server or Windows Azure SQL Database. Alternatively, the DBA can use the DACPAC to upgrade a previously deployed database using SQL Server Management Studio. To complete the lifecycle, the DBA can extract the database into a DACPAC and hand it off to a developer to either reflect test or production adjustments, or to enable further database design changes in response to changes in the application.

The advantage of a DAC-driven deployment over a script driven exercise is that the tool helps the DBA with identifying and validating behaviors from different source and target databases. During upgrades, the tool warns the DBA if the upgrade might cause data loss, and also provide an upgrade plan. The DBA can evaluate the plan and then utilize the tool to proceed with the upgrade.

DAC’s also support versioning to help the developer and the DBA maintain and manage the database lineage through its lifecycle.

[DAC Concepts](javascript:void(0))

A DAC simplifies the development, deployment, and management of data-tier elements that support an application:

* A data-tier application (DAC) is a logical database management entity that defines all SQL Server objects - such as tables, views, and instance objects – associated with a user’s database. It is a self-contained unit of SQL Server database deployment that enables data-tier developers and DBAs to package SQL Server objects into a portable artifact called a DAC package, or .dacpac file.
* For a SQL Server database to be treated as a DAC, it must be registered – either explicitly by a user operation, or implicitly by one of the DAC operations. When a database is registered, the DAC version and other properties are recorded as part of the metadata of the database. Conversely, a database can also be unregistered and have its DAC properties removed.
* In general, DAC tools are capable of reading DACPAC files generated by DAC tools from previous SQL Server versions, and can also deploy DACPAC’s to previous versions of SQL Server. However, DAC tools from earlier versions cannot read DACPAC files generated by DAC tools from later versions. Specifically:
  + DAC operations were introduced in SQL Server 2008 R2. In addition to SQL Server 2008 R2 databases, the tools support generation of DACPAC files from SQL Server 2008, SQL Server 2005 and SQL Server 2000 databases.
  + In addition to SQL Server 2012 databases, the tools shipped with SQL Server 2012 can read DACPAC files generated by DAC tools shipped with SQL Server 2008 R2. This includes databases from SQL Server 2008 R2, SQL Server 2008, and SQL Server 2005, but not SQL Server 2000.
  + DAC tools from SQL Server 2008 R2 cannot read DACPAC files generated by tools from SQL Server 2012.
* A DACPAC is a Windows file with a .dacpac extension. The file supports an open format consisting of multiple XML sections representing details of the DACPAC origin, the objects in the database, and other characteristics. An advanced user can unpack the file using the DacUnpack.exe utility that ships with the product to inspect each section more closely.
* The user must be a member of the dbmanager role or assigned CREATE DATABASE permissions to create a database, including creating a database by deploying a DAC package. The user must be a member of the dbmanager role, or have been assigned DROP DATABASE permissions to drop a database.

[DAC Tools](javascript:void(0))

A DACPAC can be seamlessly used across multiple tools that ship with SQL Server 2012. These tools address the requirements of different user personas using a DACPAC as the unit of interoperability.

* Application Developer
  + A database developer can use a SQL Server Data Tools database project to design a database. A successful build of this project results in the generation of a DACPAC contained in a .dacpac file.
  + In addition, the developer can import a DACPAC into a database project and continue to design the database.
  + SQL Server Data Tools also supports a Local DB for unconnected, client-side database application development. The developer can take a snapshot of this local database to create DACPAC contained in a .dacpac file.
  + Independently, the developer can publish a database project directly to a database without even generating a DACPAC. The publish operation follows similar behavior as the deploy operation from other tools.
* Database Administrator
  + A DBA can use SQL Server Management Studio to extract a DACPAC from an existing database, and also perform other DAC operations.
  + In addition, the DBA for a SQL Database can use the Management Portal for SQL Azure for DAC operations.
* Independent Software Vendor
  + Hosting services and other data management products for SQL Server can use the DACFx API for DAC operations.
* IT Administrator
  + IT systems integrators and administrators can use the SqlPackage.exe command line tool for DAC operations.

[DAC Operations](javascript:void(0))

A DAC supports the following operations:

* **EXTRACT** – the user can extract a database into a DACPAC.
* **DEPLOY** – the user can deploy a DACPAC to a host server. When the deployment is done from a manageability tool like SQL Server Management Studio or the Management Portal for SQL Azure, the resulting database in the host server is implicitly registered as a data-tier application.
* **REGISTER** – the user can register a database as a data-tier application.
* **UNREGISTER** – a database previously registered as a DAC can be unregistered.
* **UPGRADE** – a database can be upgraded using a DACPAC. Upgrade is supported even on databases that are not previously registered as data-tier applications, but as a consequence of the upgrade, the database will be implicitly registered.

[Backup Package (.bacpac)](javascript:void(0))

A BACPAC is an artifact that encapsulates the database schema as well as the data stored in the database. The BACPAC is a Windows file with a .bacpac extension. Similar to the DACPAC, the BACPAC file format is open – the schema contents of the BACPAC are identical to that of the DACPAC. The data is stored in JSON format.

DACPAC and BACPAC are similar but they target different scenarios. A DACPAC is focused on capturing and deploying schema, including upgrading an existing database. The primary use case for a DACPAC is to deploy a tightly defined schema to development, test, and then production environments, and the reverse: capturing production’s schema and applying it to back to test and development environments.

A BACPAC, on the other hand, is focused on capturing schema and data. A BACPAC is the logical equivalent of a database backup and cannot be used to upgrade existing databases. The primary use case for a BACPAC is to move a database from one server to another - or from a local server to the cloud - and archiving an existing database in an open format.

A BACPAC supports two main operations:

* **EXPORT**– The user can export the schema and the data of a database to a BACPAC.
* **IMPORT** – The user can import the schema and the data into a new database in the host server.

Both these capabilities are supported by the database management tools: Server Management Studio, the Management Portal for SQL Azure, and the DACFx API.

[Permissions](javascript:void(0))

You must be a member of the dbmanager role or assigned CREATE DATABASE permissions to create a database, including creating a database by deploying a DAC package. You must be a member of the dbmanager role, or have been assigned DROP DATABASE permissions to drop a database.

[Data-tier Application Tasks](javascript:void(0))

|  |  |
| --- | --- |
| **Task Description** | **Topic** |
| Describes how to use a DAC package file to create a new DAC instance. | [Deploy a Data-tier Application](http://technet.microsoft.com/en-us/library/ee210569.aspx) |
| Describes how to use a new DAC package file to upgrade an instance to a new version of the DAC. | [Upgrade a Data-tier Application](http://technet.microsoft.com/en-us/library/ee634742.aspx) |
| Describes how to remove a DAC instance. You can choose to also detach or drop the associated database, or leave the database intact. | [Delete a Data-tier Application](http://technet.microsoft.com/en-us/library/ee240822.aspx) |
| Describes how to view the health of currently deployed DACs by using the SQL Server Utility. | [Monitor Data-tier Applications](http://technet.microsoft.com/en-us/library/ee210574.aspx) |
| Describes how to create a .bacpac file that contains an archive of the data and metadata in a DAC. | [Export a Data-tier Application](http://technet.microsoft.com/en-us/library/hh213241.aspx) |
| Describes how to use a DAC archive file (.bacpac) to either perform a logical restore of a DAC, or to migrate the DAC to another instance of the Database Engine or SQL Database. | [Import a BACPAC File to Create a New User Database](http://technet.microsoft.com/en-us/library/hh710052.aspx) |
| Describes how to import a BACPAC file to create a new user database within an instance of SQL Server. | [Extract a DAC From a Database](http://technet.microsoft.com/en-us/library/hh231291.aspx) |
| Describes how to promote an existing database to be a DAC instance. A DAC definition is built and stored in the system databases. | [Register a Database As a DAC](http://technet.microsoft.com/en-us/library/ee633653.aspx) |
| Describes how to review the contents of a DAC package and the actions a DAC upgrade will perform before using the package in a production system. | [Validate a DAC Package](http://technet.microsoft.com/en-us/library/ee633948.aspx) |
| Describes how to place the contents of a DAC package into a folder where a database administrator can review what the DAC does before deploying it to a production server. | [Unpack a DAC Package](http://technet.microsoft.com/en-us/library/ee633768.aspx) |
| Describes how to use a wizard to deploy an existing database. The wizard uses DACs to perform the deployment. | [Deploy a Database By Using a DAC](http://technet.microsoft.com/en-us/library/jj554810.aspx) |

# Monitor Data-tier Applications

**SQL Server 2012**

[Other Versions](javascript:;)



* [SQL Server 2008 R2](http://technet.microsoft.com/en-us/library/ee210574(d=printer,v=sql.105).aspx)

A data-tier application (DAC) can be monitored from the Utility Explorer and Object Explorer in SQL Server Management Studio (SSMS), along with system views and tables. In addition, all objects in the database contained in the DAC can be monitored using standard database and Database Engine monitoring techniques.

[Before You Begin](javascript:void(0))

If you deploy a DAC to a managed instance of the Database Engine, information about the deployed DAC is incorporated into the SQL Server Utility the next time the utility collection set is sent from the instance to the utility control point. You can then view basic health information about the DAC by using the Management Studio Utility Explorer.

The SSMS Object Explorer displays basic configuration information about each DAC deployed to an instance of the Database Engine, regardless of whether the instance is managed in the SQL Server Utility. Also, the database associated with a deployed DAC can be monitored using the same procedures for monitoring any database.

[Using the SQL Server Utility](javascript:void(0))

The Deployed Data-tier Applications detail page in the Management Studio Utility Explorer displays a dashboard that reports the resource utilization of all DACs that have been deployed to managed instances of the Database Engine. The top pane of the details page lists each deployed DAC with visual indicators showing whether their utilization of CPU and file resources are outside the policies defined for the SQL Server Utility. If you select any DAC in the list view, further details are displayed in tabs in the bottom pane of the page. For more information about the information presented on the details page, see [Deployed Data-tier Application Details (SQL Server Utility)](http://technet.microsoft.com/en-us/library/ee240857.aspx).

After using the Deployed Data-tier Applications detail page to quickly identify any DACs that are either under-utilizing or stressing their hardware resource, you can make plans to address any problems. Multiple DACs that are not fully utilizing their current hardware resources could be consolidated to a single server, freeing some of the servers for other uses. If a DAC is stressing the resources on its current server, the DAC can be moved to a larger server, or additional resources can be added to the current server.

The minimum and maximum limits for resource usage are defined by application monitoring policies defined in the Utility Administration details page. Database administrators can tailor the policies to match the limits established by their organizations. For example, one company might set 75% as the maximum CPU utilization for a DAC, while another company might set the maximum at 80%. For more information about setting application monitoring policies, see [Utility Administration (SQL Server Utility)](http://technet.microsoft.com/en-us/library/ee240832.aspx).

To view the Deployed Data-tier Applications detail page:

1. Select the View/Utility Explorer menu.
2. Connect the Utility Explorer to the utility control point (UCP).
3. Select the View/Utility Explorer Details menu.
4. Select the Deployed Data-tier Applications node in the Utility Explorer.

The information in the Deployed Data-tier Applications detail page comes from the data in the utility management data warehouse, which defaults to collecting the data every 15 minutes. The interval can also be tailored using the Utility Administration details page.

[Using Object Explorer](javascript:void(0))

The SSMS Object Explorer displays basic configuration information about each DAC deployed to an instance of the Database Engine. This includes both managed instances that have been enrolled in the SQL Server Utility, and stand-alone instances that cannot be viewed in the Utility Explorer.

To view the details of a DAC deployed to an instance of the Database Engine:

1. Select the View/Object Explorer menu.
2. Connect to the instance of the from the Object Explorer pane.
3. Select the View/Object Explorer Details menu.
4. Select the server node in Object Explorer that maps to the instance, and then navigate to the Management\Data-tier Applications node.
5. The list view in the top pane of the details page lists each DAC deployed to the instance of the Database Engine. Select a DAC to display the information in the detail pane at the bottom of the page.

The right-click menu of the Data-tier Applications node is also used to deploy a new DAC or delete an existing DAC.

[Using the DAC System Views and Tables](javascript:void(0))

The msdb.dbo.sysdac\_history\_internal system table records the success or failure of all DAC management actions performed on an instance of the Database Engine. The table records the time each action occurred, and which login initiated the action. For more information, see [sysdac\_history\_internal (Transact-SQL)](http://technet.microsoft.com/en-us/library/ee210551.aspx).

The DAC system views report basic catalog information. For more information, see [Data-tier Application Views (Transact-SQL)](http://technet.microsoft.com/en-us/library/ee240740.aspx).

[Monitoring DAC Databases](javascript:void(0))

After a DAC has been successfully deployed, the database contained in the DAC operates the same as any other database. Use standard techniques and tools for monitoring the performance, log, events, and resource utilization of the database.