Microsoft

SQL Server Technical Article

Writers: Joanne Hodgins, Omri Bahat, Morgan Oslake, and Matt Hollingsworth

Technical Reviewer: Dan Jones

Published: August 2009

Applies to: SQL Server 2008 R2, August CTP

Summary:

SQL Server 2008 R2 introduces new management tools to help improve IT efficiency and productivity. Investments in application and multi-server management will help organizations proactively manage database environments efficiently at scale through centralized visibility into resource utilization. Such investments can help streamline consolidation and upgrade initiatives across the application lifecycle—all with tools that make it fast and easy.

This paper introduces the new extensions in SQL Server Management Studio and the Control Point Explorer, and it walks through the simple process of setting up a SQL Server managed server group, including SQL Server Control Point installation, enrolling an instance into central management, extracting Data-tier Applications from existing deployments, and deploying Data-tier Applications to the new managed server group.

Copyright

This is a preliminary document and may be changed substantially prior to final commercial release of the software described herein.

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This white paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Contents

Introduction

Microsoft's investments in application and multi-server management will help organizations manage database environments more efficiently at scale with visibility into resource utilization for consolidation and improved efficiencies across the application lifecycle. A core concept to application and multi-server management is the addition of the Control Point Explorer, which enables a centralized view of Microsoft® SQL Server® instances and database applications and their utilization across the designated managed server group.

What's more, for centralized SQL Server management to provide incremental value, database administrators (DBAs) need a single unit of deployment for their database applications to accelerate deployments, moves, and upgrades. This is especially important for streamlining the tasks associated with consolidation management. To this end, SQL Server 2008 R2 introduces a new concept, the Data-tier Application. A Data-tier Application is a container that defines and bundles database schema and deployment requirements of an application.

This paper introduces the new extensions in SQL Server Management Studio, the Control Point Explorer, and the new single unit of deployment concept, the Data-tier Application. The paper walks through the simple process of setting up a SQL Server managed server group, including setting up a SQL Server Control Point, enrolling an instance into central management, extracting Data-tier Applications from existing deployments, and deploying Data-tier Applications to the new managed server group.

New Terms

Control Point Explorer – Accessed from SQL Server Management Studio, the Control Point Explorer serves as the entry point to many of the application and multi-server management enhancements. This component is currently named "Utility Explorer" in the August Community Technology Preview (CTP).

SQL Server Control Point – Accessed via the Control Point Explorer, a SQL Server control point is a SQL Server instance designated to maintain relationships with enrolled SQL Server instances within a managed server group. This component is currently named "Utility Control Point" or "UCP" in the August CTP.

Managed Server Group – Describes a group of SQL Server instances enrolled into a SQL Server control point, where utilization data is collected and accessible through the dashboard viewpoints in the SQL Server control point.

Data-tier Application Component (DAC) – Interoperability with the Microsoft Visual Studio® development system introduces a new project template called Data-tier Application Component (DAC). This project template captures the database application schema (tables, stored procedures, and so forth) and packages it with application deployment requirements, enabling a single unit of deployment. The DAC serves as the file read by the new wizards within the Control Point Explorer that will unpack the application schema and deployment requirements for deployment as the Data-tier Application.

Data-tier Application – the Data-tier Application is an unpacked DAC file deployed on an enrolled SQL Server instance within a managed server group. The deployed Data-tier Application is connected with the SQL Server Control Point enabling utilization data to be collected and capacity policies evaluated.

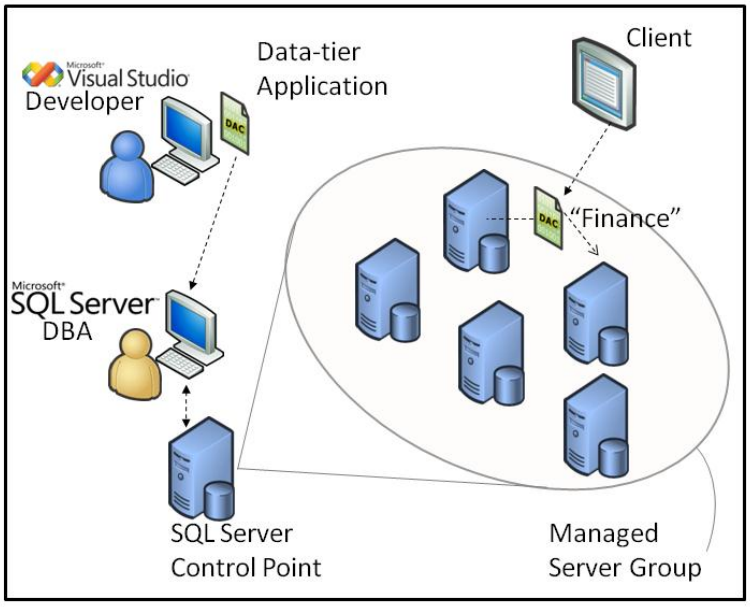

Figure 1: Application and Multi-Server Management concepts

Introducing the SQL Server Managed Server Group

Customers have an increasingly important requirement to manage their SQL Server environment as a whole, focusing more on managing all of their Data-tier Applications and less on managing individual computers and instances of SQL Server. SQL Server 2008 R2 addresses this requirement through the concept of the SQL Server managed server group. This models an organization's SQL Server-related entities in a unified view. Entities that can be managed include instances of SQL Server, data-tier applications, database files, processor utilization, and storage utilization. This new way to organize and monitor SQL Server resource capacity enables administrators to have a holistic view of their environment.

The managed server group is managed through a SQL Server control point (SCP) using the new Control Point Explorer in SQL Server Management Studio (SSMS). The SCP is configured on a SQL Server instance and provides the central reasoning point for a managed server group. It contains configuration and performance information collected by managed instances of SQL Server, and it stores this information in a central management repository. SQL Server configuration settings and performance data are collected and then compared to policy evaluation results on the SCP to help administrators identify resource utilization bottlenecks and

consolidation opportunities. The SCP also contains data used for impact analysis and what-if scenarios.

The SQL Server managed server group model contains three layers:

- **Data-tier Applications**—the data-tier applications managed by the organization.
- **SQL Server Runtimes**—the instances of the Database Engine used by the organization.
- **Hardware Resources**—the resources used by the SQL Server Runtimes, like computers and disk storage systems.

A Data-tier Application is a container that defines and bundles the database schema, reference data, and deployment requirements of an application. The Data-tier Application forms a file that enables a single unit of deployment, for the full lifecycle of an application, including versioning. It further enables data-tier automation by providing a means to capture the intent of the developer and deployment-specific details. It abstracts the application data-tier by providing well-known endpoint names instead of computer and instance names, so a data-tier application can be moved between SQL Server runtimes without requiring application changes.

The managed server group will support actions like creating a SQL Server control point and declaring policies that establish utilization thresholds then deploying them to specific SQL Server runtimes whose properties comply with the server selection policies defined in the Data-tier Application.

The Control Point Explorer within the SQL Server Management Studio user interface provides a hierarchical tree view, similar to the SQL Server Management Studio Object Explorer, for navigating through and managing the entities in the SQL Server managed server group. Viewpoints and dashboards provide views into the capacity of the elements in the managed server group.

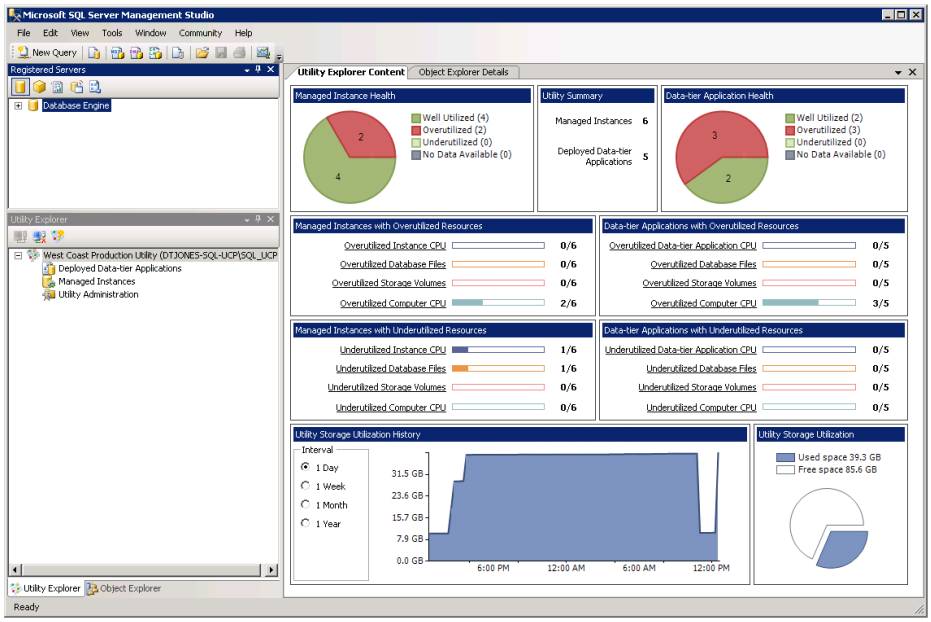

Figure 2: Dashboard viewpoints of instance and application utilizations

Creating a SQL Server Managed Server Group

This section will walk through setting up a SQL Server instance as a SQL Server control point, explain new concepts and terminology associated with the Control Point Explorer and a SQL Server control point, and walk through enrolling a SQL Server instance into the managed server group for insights into resource utilization.

Setting Up a SQL Server Control Point

In order to create the SQL Server control point, you can invoke the Create Control Point Wizard in SQL Server Management Studio. On the **View** menu in SQL Server Management Studio, click **Utility Explorer**, and then click the **Create UCP** icon.

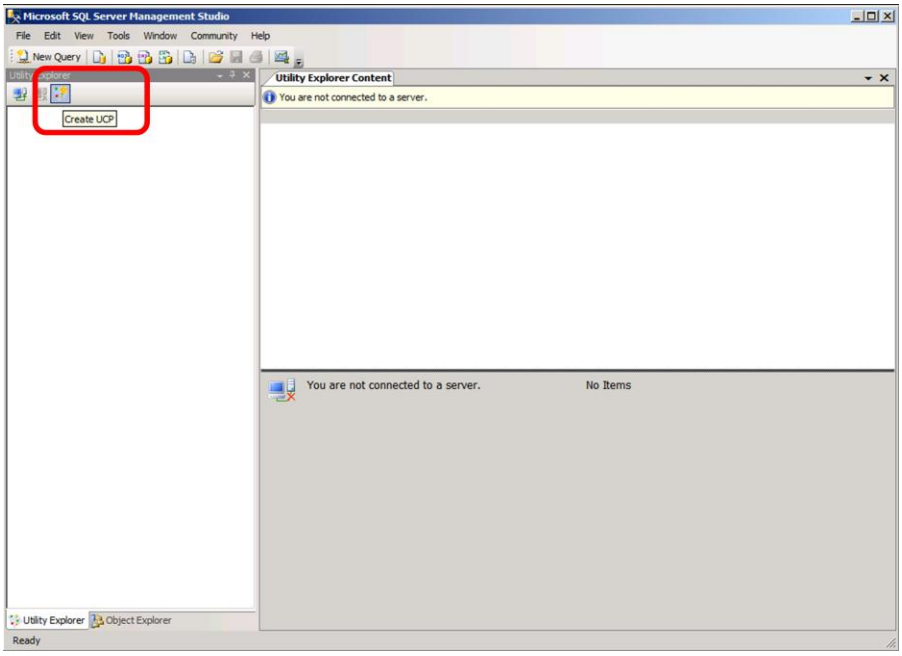

Figure 3: The location in SQL Server Management Studio where the Create Control Point Wizard is launched

This launches the Create Control Point Wizard, which begins with an introduction page. The wizard creates a control point on an instance of SQL Server. This creation process includes provisioning the control point schema, jobs, and polices as well as creating a management data warehouse.

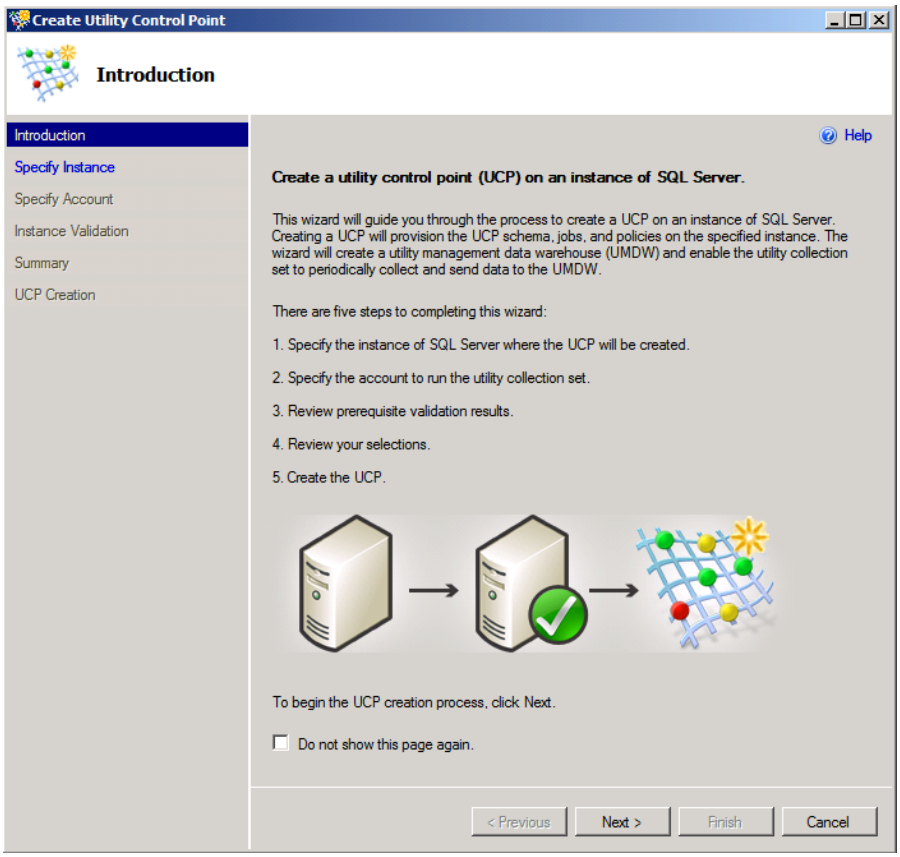

Figure 4: The Introduction page of the Create Control Point Wizard

The Specify Instance page of the Create Control Point Wizard asks you to identify the SQL Server instance that will become the control point. You are also able to provide a friendly name for the control point in this page.

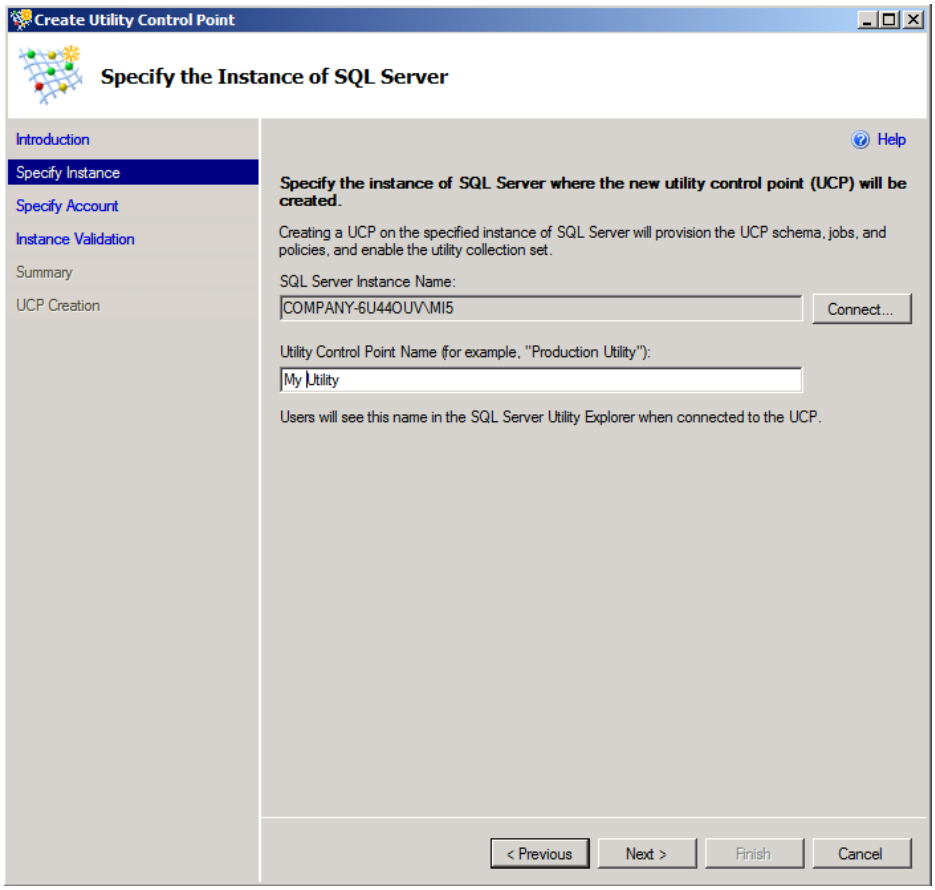

Figure 5: The Specify Instance page of the Create Control Point Wizard

The Specify Account page of the Create Control Point Wizard asks you to identify the Windows® domain account to be used as the SQL Server Agent account for the managed server group. This is the account for the collection set that is run on the control point instance itself.

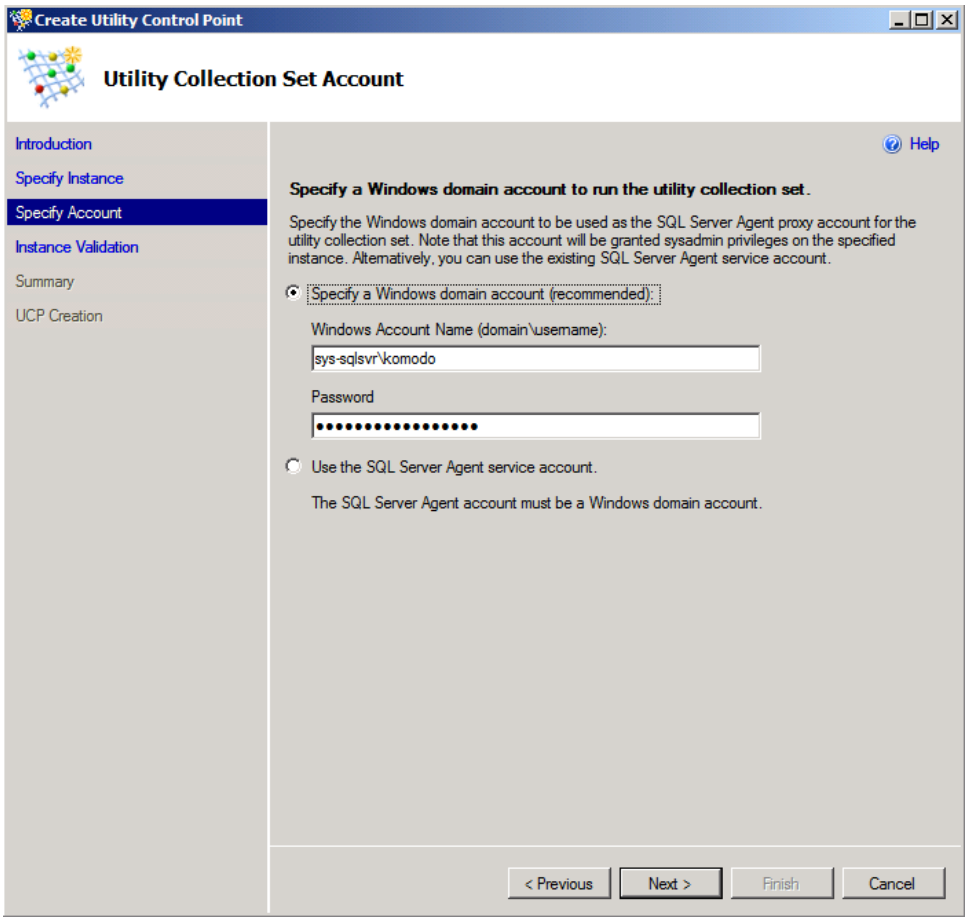

Figure 6: The Specify Account page of the Create Control Point Wizard

The Instance Validation page of the Create Control Point Wizard verifies that conditions required to create a Control Point on the specified instance are satisfied.

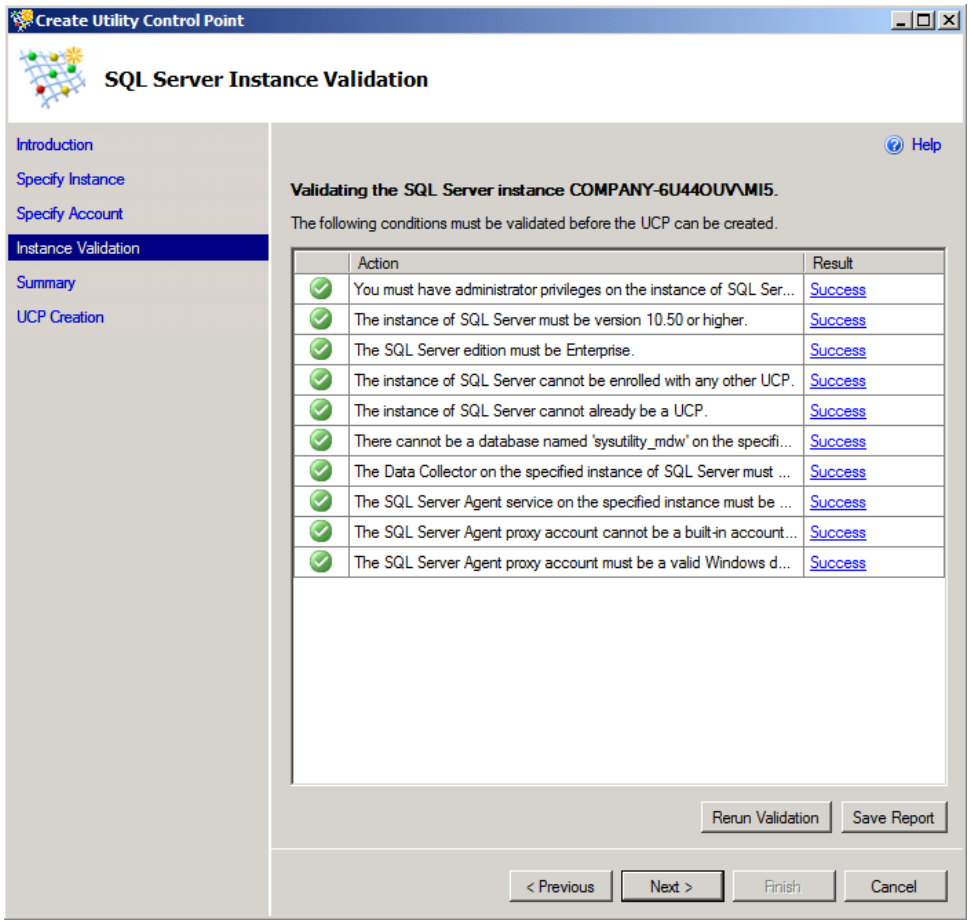

Figure 7: The Instance Validation page of the Create Control Point Wizard

The Summary page of the Create Control Point Wizard displays the choices made in previous pages of the wizard.

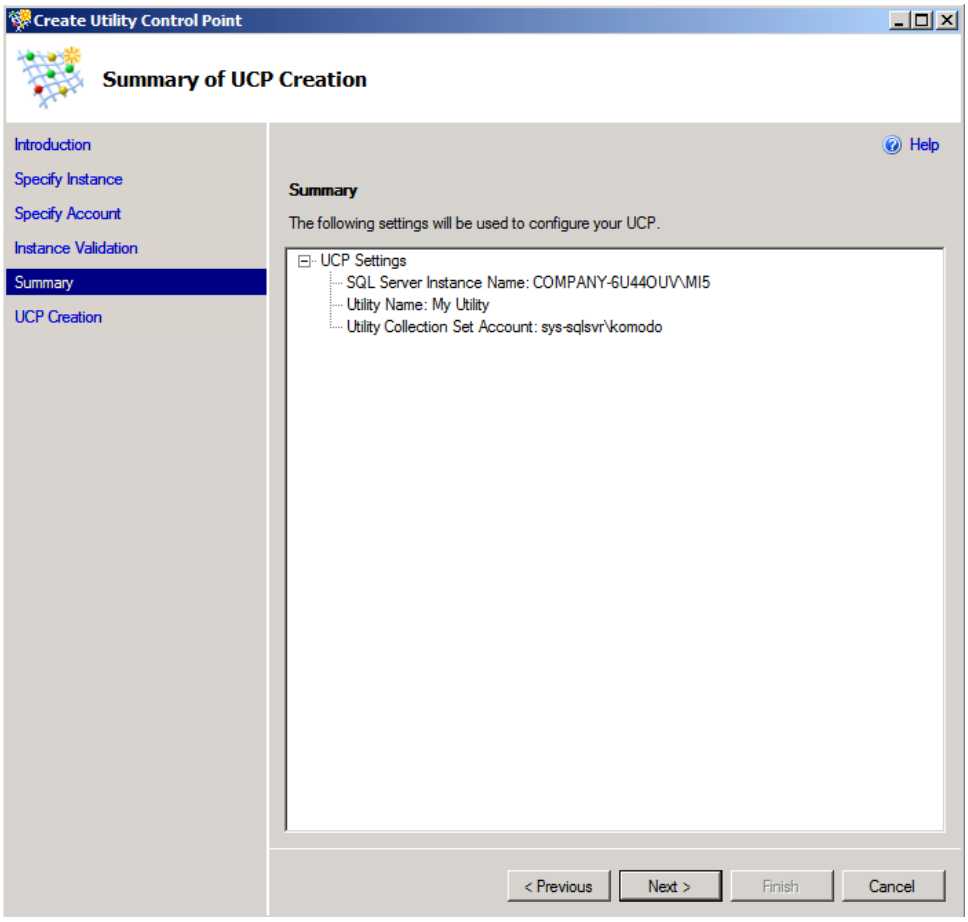

Figure 8: The Summary page of the Create Control Point Wizard

The Control Point Creation page of the Create Control Point Wizard shows the steps that executed in order to create the control point.

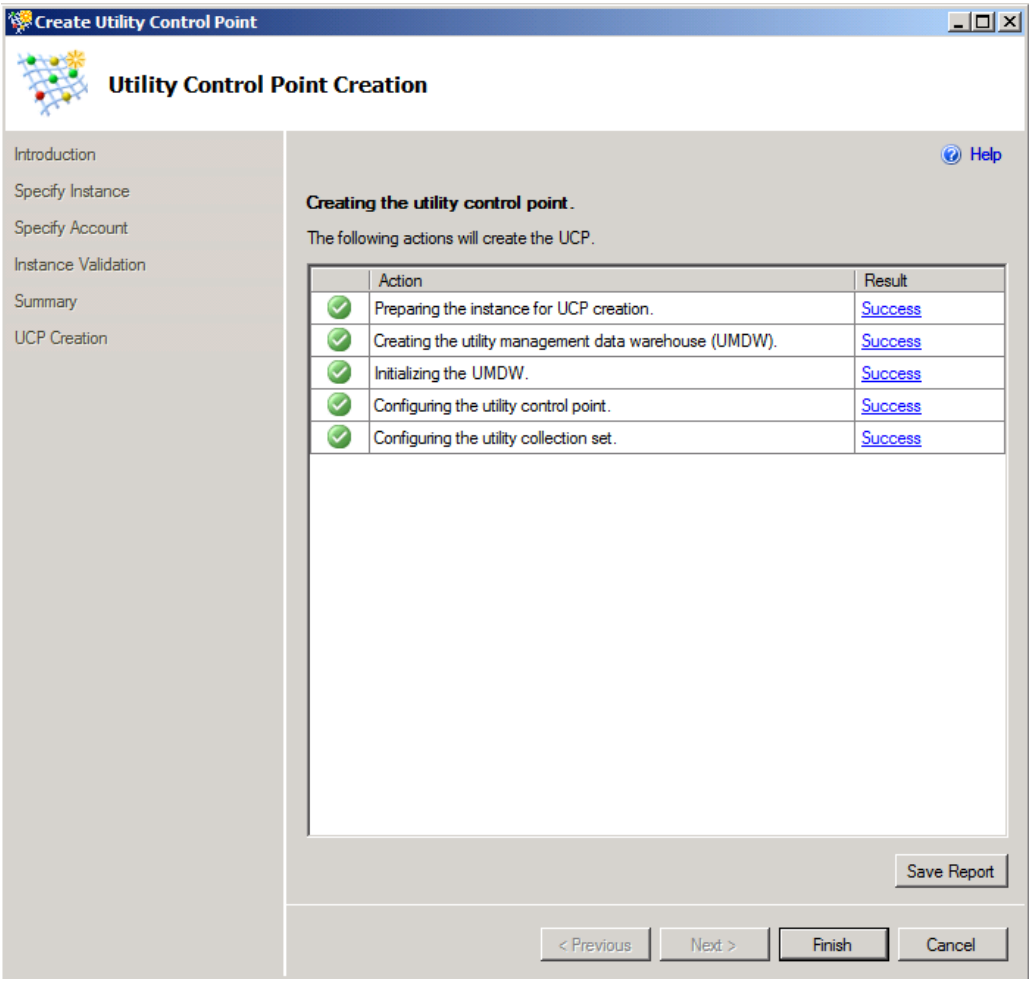

Figure 9: The Creation page of the Create Control Point Wizard

Enroll an Instance into the SQL Server Control Point

After the control point is created, you can enroll instances that you want to manage.

In order to enroll an instance in the control Point, you can invoke the Enroll Instance Wizard from the Control Point Explorer (currently named Utility Explorer) pane in SQL Server Management Studio. In the Control Point Explorer, right-click **Managed Instances**, and then click **Add Managed Instance**.

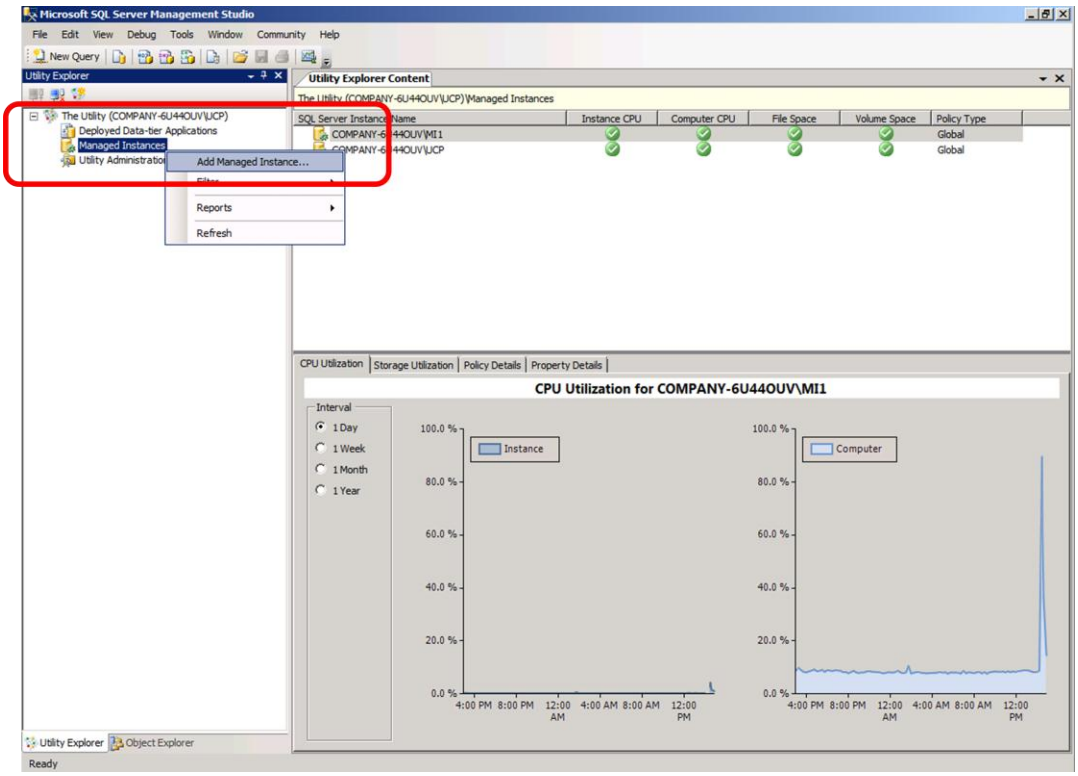

Figure 10: Where the Enroll Instance Wizard is launched from within the Control Point Explorer

This launches the Enroll Instance Wizard, which begins with an introduction page. The Enroll Instance Wizard enrolls a SQL Server instance as a managed instance in the control point. This enrollment process will start the managed server group collection set, which will upload data to the control point once every 15 minutes.

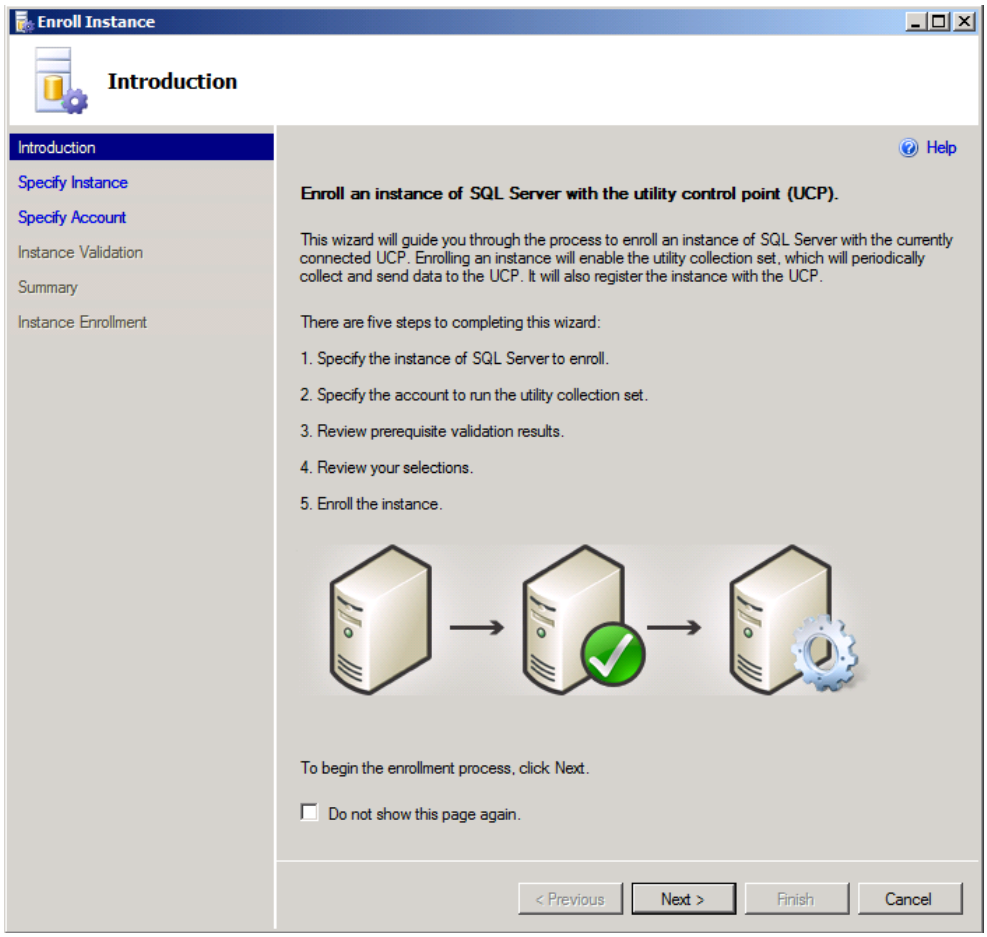

Figure 11: The Introduction page of the Enroll Instance Wizard

The Specify Instance page of the Enroll Instance Wizard asks identifies the SQL Server instance that will become a managed instance of the control point.

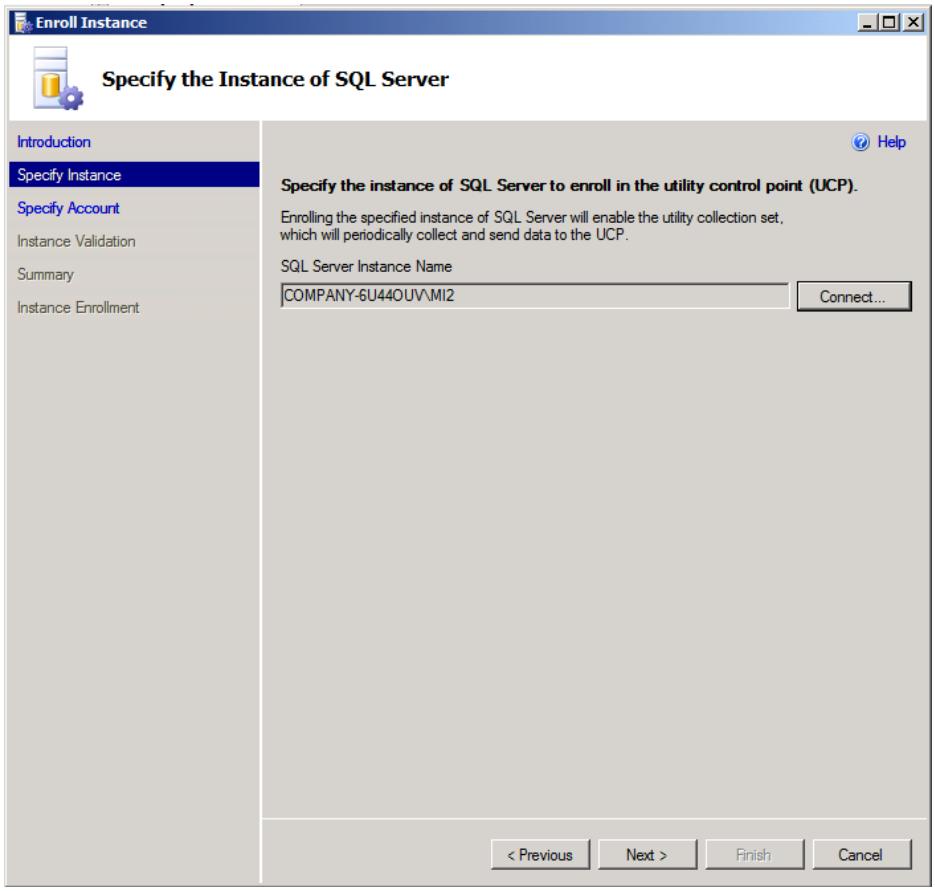

Figure 12: The Specify Instance page of the Enroll Instance Wizard

The Specify Account page of the Enroll Instance Wizard asks you to identify the Windows domain account to be used as the SQL Server Agent account for the managed server group collection set. This is the account for the collection set that is run on the managed instance.

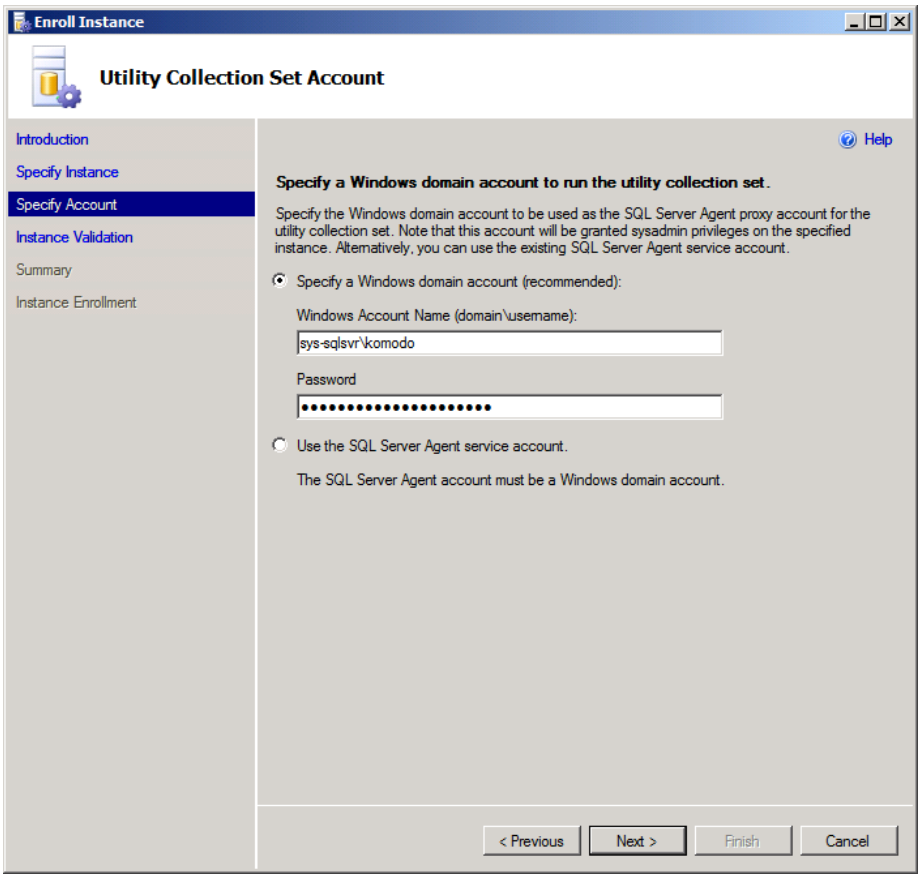

Figure 13: The Specify Account page of the Enroll Instance Wizard

The Instance Validation page of the Enroll Instance Wizard verifies that conditions required to enroll the specified instance as a managed instance are satisfied.

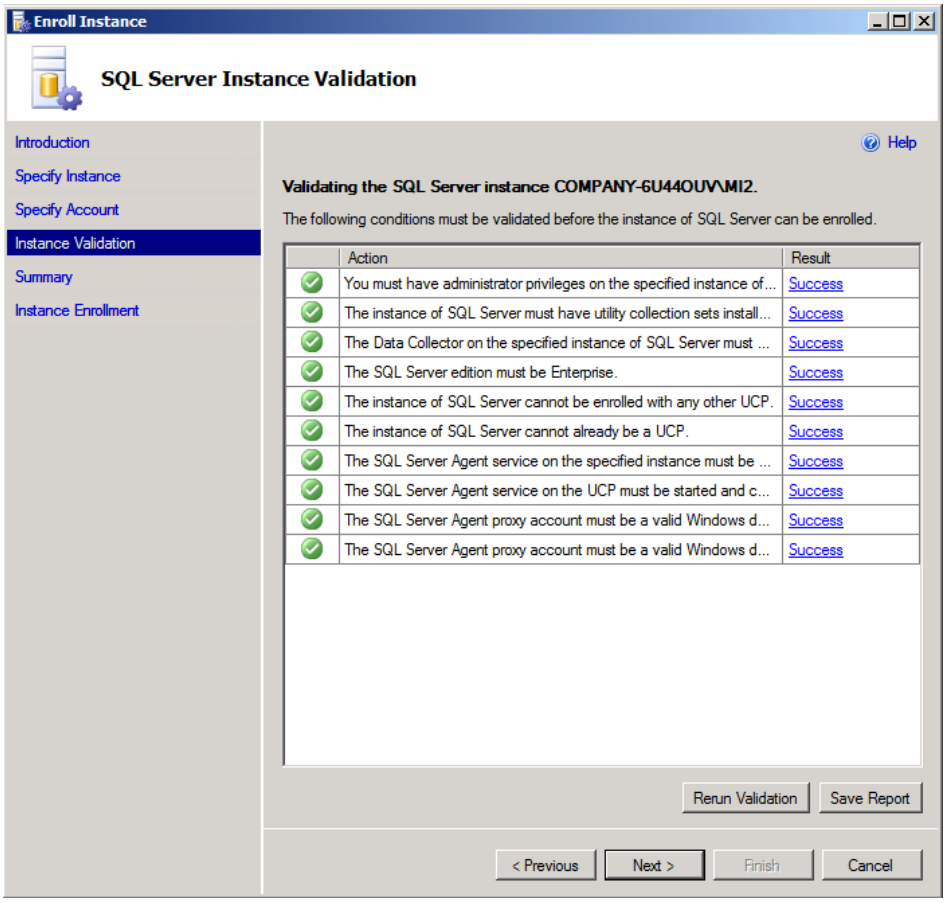

Figure 14: The Instance Validation page of the Enroll Instance Wizard

The Summary page of the Enroll Instance Wizard displays the choices made in previous pages of the wizard.

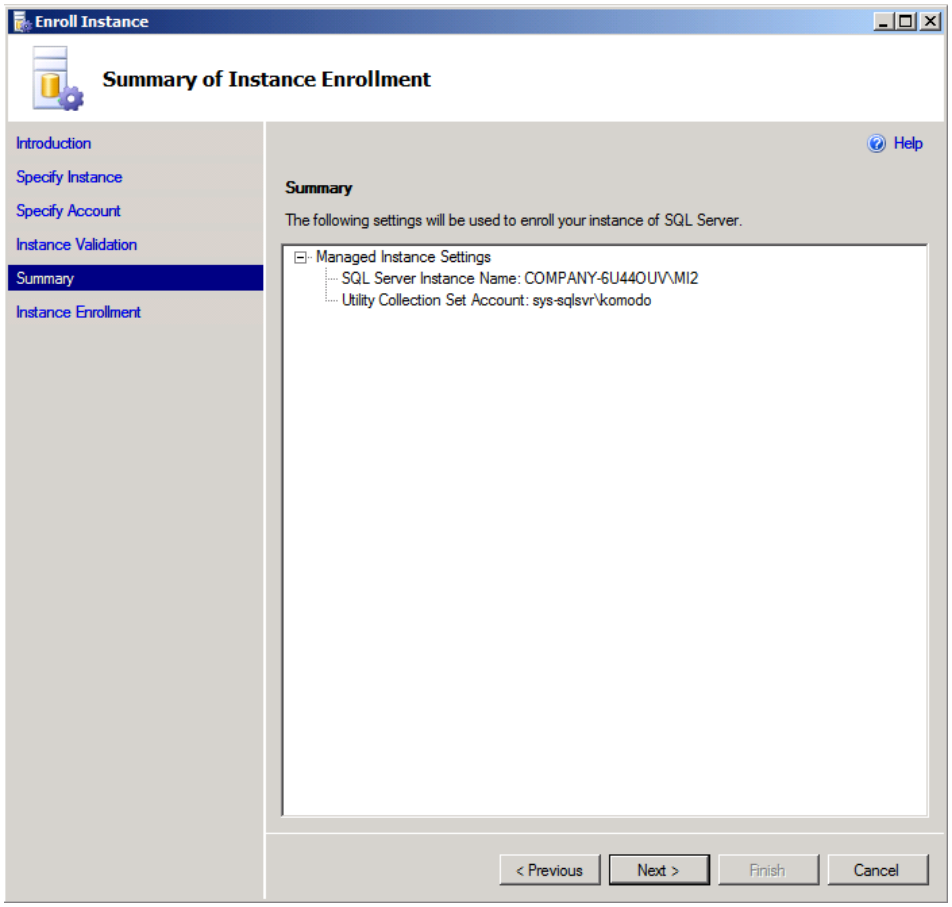

Figure 15: The Summary page of the Enroll Instance Wizard

The Enroll Instance page of the Enroll Instance Wizard shows the steps that executed in order to enroll the instance.

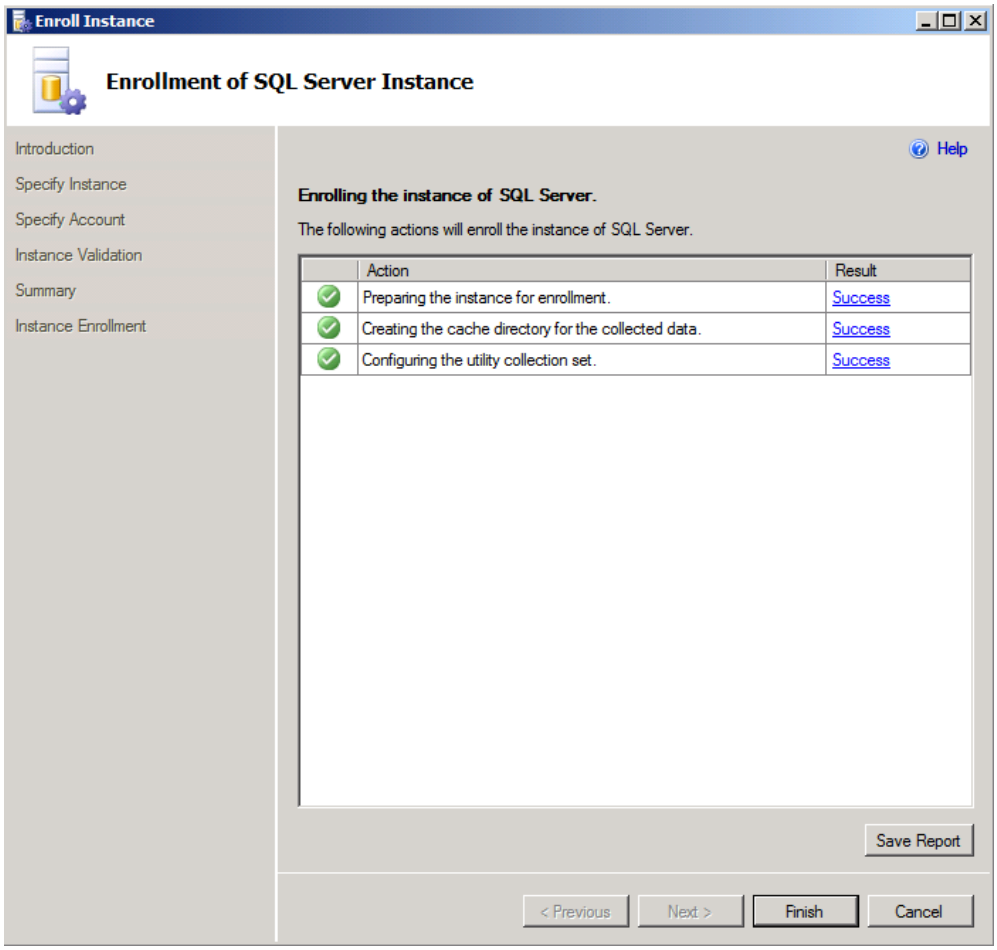

Figure 16: The Instance Enrollment page of the Enroll Instance Wizard

Understanding the Data within the SQL Server Control Point

Summary view – Pie charts at the top of the dashboard provide at-a-glance summaries of resource health for managed SQL Server instances and data-tier applications. The summary at the top-center of the dashboard displays the total numbers of managed SQL Server instances and data-tier application components in the managed server group.

In the dashboard summary for managed instance health, a SQL Server instance is marked as overutilized if any of the following conditions are true:

- CPU resources for the instance of SQL Server are overutilized.
- CPU resources of the computer that hosts the SQL Server instance are overutilized.
- The instance contains data or log files with overutilized storage space.
- The instance contains data or log files that reside on volumes with overutilized storage space.

In the dashboard summary for managed instance health, a SQL Server instance is marked as underutilized if it is not marked as overutilized and any of the following conditions are true:

- CPU resources allocated to the instance of SQL Server are underutilized.

- CPU resources of the computer that hosts the SQL Server instance are underutilized.
- The instance contains data or log files with underutilized storage space.
- The instance contains data or log files that reside on volumes with underutilized storage space.

In the dashboard summary for managed instance health, a SQL Server instance is marked as well utilized if it is not marked as overutilized and it not marked as underutilized.

Similar rules are used in the dashboard summary for data tier application health.

Rollup view - Sliding gauges below the pie charts show a summary of the number of managed instances of SQL Server and data-tier applications for each resource utilization dimension – for example, CPU utilization for instances of SQL Server, CPU utilization for the entire computer, file space utilization, and storage volume space utilization.

Storage Utilization view – Graphic representations at the bottom of the dashboard show an aggregation of current utilization and utilization history of disk space use of storage resources across the entire managed server group.

Detail view – To view detailed information about resource utilization for specific computers, instances of SQL Server, and Data-tier Applications, navigate to the Control Point Explorer navigation pane and then click **Managed Instances** or **Deployed Data-tier Applications**. The detailed list view displays the health states for data-tier applications and SQL Server instances across key resource utilization dimensions.

These dimensions include processor utilization and storage space utilization. The health states in the viewpoints represent either underutilized resources marked with a green down arrow icon, overutilized resources marked with a red up arrow icon, or resources that are neither underutilized nor overutilized marked with a green check icon. The underutilization and overutilization of a given resource are defined by resource utilization policies. The underutilization policy defines the underutilization threshold, and the overutilization policy defines the overutilization threshold – these policies have default settings that are easily adjustable with slider bars.

The list view also has tabs with details about processor utilization, storage space utilization, and deployment properties for every managed instance of SQL Server and data-tier application in the managed server group.

Data-tier Application Overview

To simplify the development and deployment of the data-tier, Microsoft has introduced the concept of a data-tier application. A Data-tier Application is a single unit of deployment that captures data objects and data-tier application artifacts. In other words, it is a container that includes server and database schema objects that are used by an application (for example., tables, views, logins and users), as well as deployment prerequisites that set the requirements on the instances of SQL Server where Data-tier Applications can be deployed. The output file

for a Data-tier Application is a Data-tier Application Component (.dacpac), this file is unpacked and deployed to a managed instance.

The key benefit of using DACs is that they are designed to offer data-tier automation, including:

- Collecting large numbers of entities into one DAC that can be managed as a single unit through the full lifecycle of an application, including versioning.
- Automating the lifecycle of the data-tier by enabling developers to make changes to a data-tier application component, package it, and then pass it to DBAs for final deployment.
- Including policies that capture the intent of the developers, as well as deployment requirements on the instances of SQL Server where DACs can be deployed.

This section is focused on extracting DACs from existing instances of SQL Server, as well as deploying a DAC to a target instance of SQL Server. With these features, users can create a corresponding DAC package file from within an existing database. The DAC package can then be opened and edited in Visual Studio 2010 to make changes to the data-tier. Visual Studio users can then build an updated DAC package file and send the changes to DBAs, who in turn deploy the updates using SQL Server Management Studio. These capabilities are expected to be available in Visual Studio 2010 in the Fall 2009 timeframe.

DAC extraction and deployment are also extremely useful for upgrading databases to the SQL Server 2008 platform. Users can move databases by extracting a DAC from an instance of SQL Server 2000, SQL Server 2005, or SQL Server 2008, and then deploy the extracted DAC to an instance of SQL Server 2008 R2, through either SQL Server Management Studio or the Windows PowerShell™ command-line interface.

Extracting and Deploying Data-tier Applications

A Data-tier Application can be extracted from an existing SQL Server database. The extraction process creates a DAC package file that contains all database objects and their related SQL Server elements. For example, a DAC package file contains all database tables, stored procedures, views, users, and logins that map to the database users. The DAC package can then be deployed to a target instance of SQL Server, to either create a new instance of the Data-tier Application or to upgrade an existing Data-tier Application.

Creating a Data-tier Application from an existing database - In order to create Data-tier Applications from databases, users can invoke the Extract Data-tier Application Wizard from the Object Explorer pane in SQL Server Management Studio. By doing so, users can launch the wizard from an existing instance and then create a DAC package file that contains the Data-tier Application corresponding to the selected database and instance.

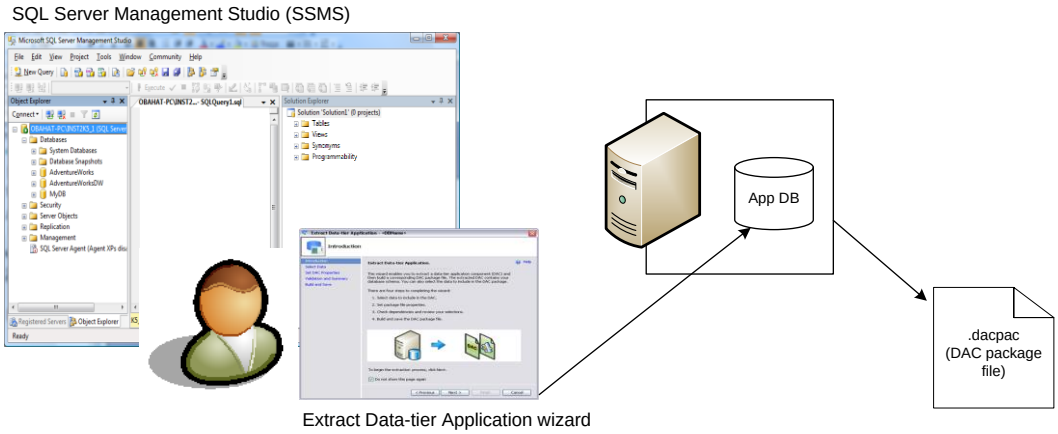

Figure 17: Extracting a DAC from an existing database.

Developing data-tier changes – After a user creates a DAC package (for example, after a DBA extracts the necessary objects from an instance of SQL Server in production), the DAC package file can be sent to developers. The developers can open the DAC package into Visual Studio, where they can update the database and instance schema in the Data-tier Application. For example, developers can create new tables, logins, and users; remove views and stored procedures; and so on. After changes are coded in the Visual Studio project system, developers *build* the project and compile a new version of the DAC to create an updated DAC package file.

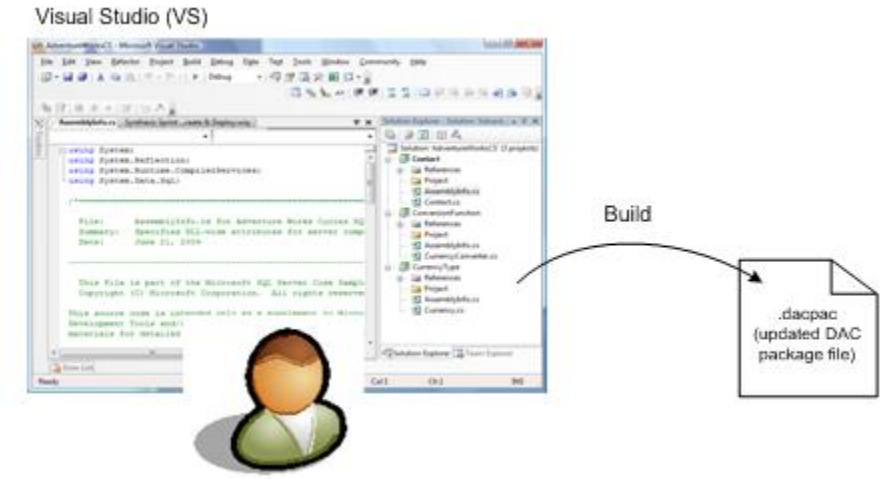

Figure 18: Making data-tier changes and creating an updated DAC package file

Moving changes to test and production – After developers create and test the new DAC package file, the file can then be sent to DBAs and change managers (that is, deploying-users) working in the test or production SQL Server environments. To push the changes to their databases and instances of SQL Server, deploying-users can activate the Deploy Data-tier Application Wizard in SQL Server Management Studio, configure the deployment parameters, and then propagate the changes to the appropriate target instance of SQL Server.

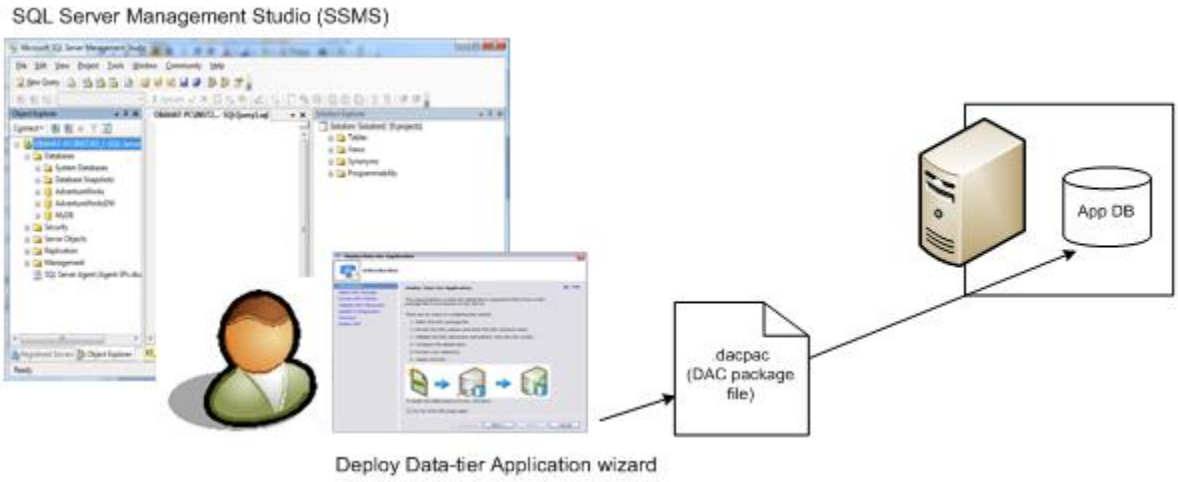

Figure 19: Deploying a DAC to an instance of SQL Server

Upgrading to SQL Server 2008 R2 – DACs enable DBAs to upgrade the schema of SQL Server databases and instances from SQL Server 2000, SQL Server 2005, and SQL Server 2008 to SQL Server 2008 R2. To upgrade, users can point to a SQL Server 2000, SQL Server 2005, or SQL Server 2008 database, extract a DAC and create the corresponding DAC package file, and then deploy the DAC to a target instance of SQL Server 2008 R2. Then, data can be transferred from the source database to the target database by using SQL Server Integration Services, the bulk copy utility, or many other data migration techniques.

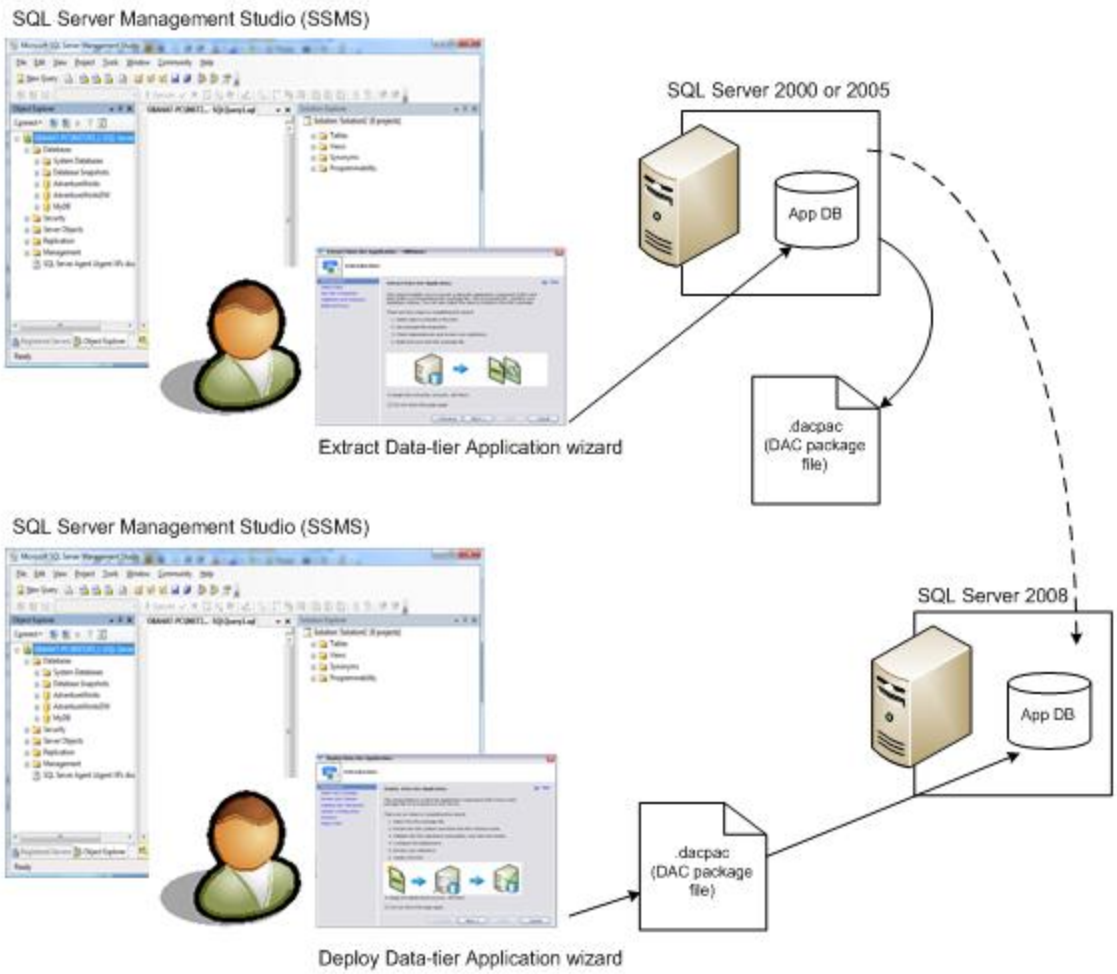

Figure 20: Upgrading to SQL Server 2008 R2 by extracting a DAC from SQL Server 2000 or SQL Server 2005 and then deploying the DAC package file to an instance of SQL Server 2008 or 2008 R2

How to Extract a Data-tier Application

The following instructions demonstrate how to extract a data-tier application from an instance of SQL Server using SQL Server Management Studio:

1. After installing a SQL Server 2008 R2 server and client tools, launch SQL Server Management Studio.
2. Connect the instances of SQL Server that you will be using during this CTP. To register an instance of SQL Server, in Object Explorer, click **Connect**, click **Database Engine**, and then follow the instructions in the **Connect to Server** dialog box.
3. In Object Explorer, under one of your registered instances, expand the Databases node and select a user-database. Next, tight-click the user database, click **Tasks**, and then click **Extract a Data-tier Application**.
4. Read the instructions in the Introduction page of the Extract Data-tier Application Wizard, and then click Next.

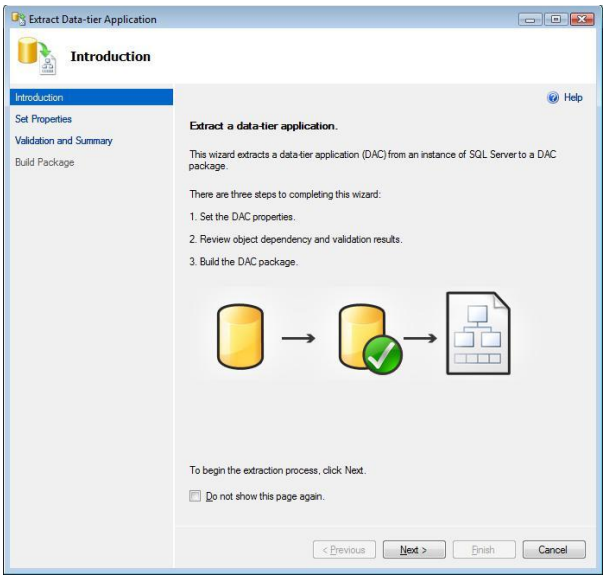

Figure 21: The Introduction page of the Extract Data-tier Application Wizard

5. In the Set Properties page of the wizard, enter the DAC name and version, as well as the file name and path for the DAC package file. Click **Next** when you are done.

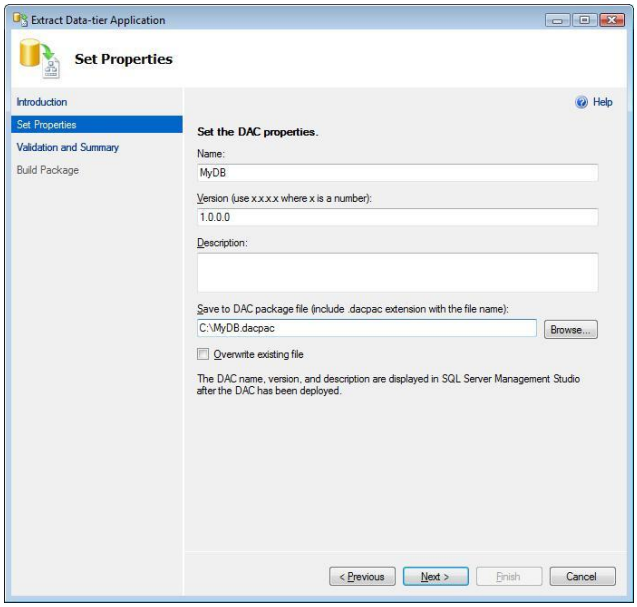

Figure 22: The Set Properties page of the Extract Data-tier Application Wizard

Next, the Validation and Summary page appears. Here, the wizard checks dependencies between database objects and verifies that all objects are supported by a DAC. After this is done, the wizard displays all findings in a summary report. If all objects are supported, click **Next** to extract the DAC. However, if some objects are not supported, **Next** is disabled, and you should stop the wizard and extract a DAC from a different database.

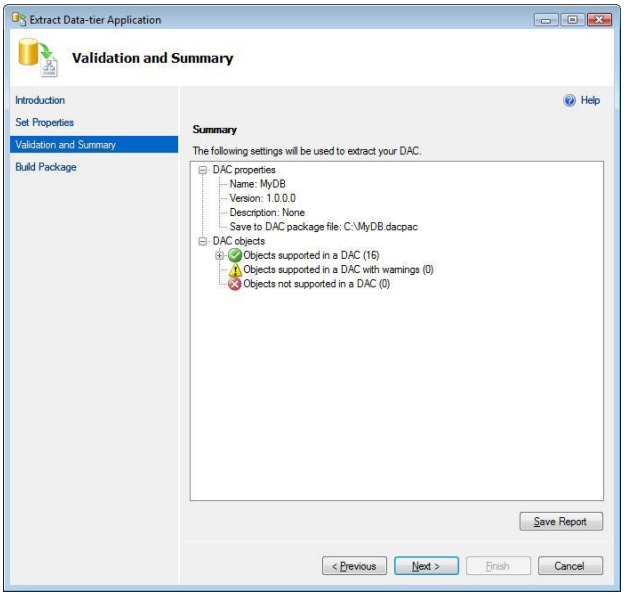

Figure 23: The Validation and Summary page of the Extract Data-tier Application Wizard

The last page of the wizard is the Build Package page, where the DAC is extracted and the DAC package file is written to the file system. When this page completes, click **Finish** to complete the wizard.

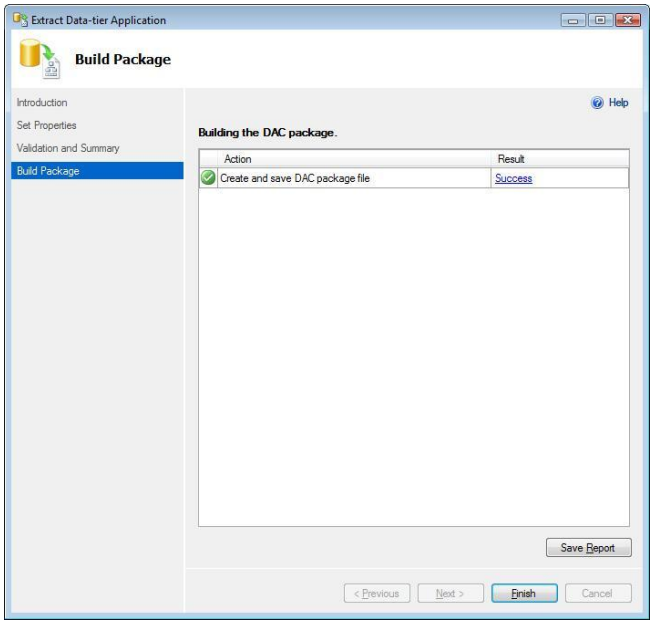

Figure 24: The Build Package page of the Extract Data-tier Application Wizard

Verify that a new DAC package file was created in the path you specified in step 5.

The DAC package file can be imported into a DAC Visual Studio project, or it can be deployed to a SQL Server instance.

How to Deploy a Data-tier Application

1. Find the DAC package file for the DAC you want to deploy. If you do not have any DAC package files, you can extract a DAC to create a package file by following the instructions in the last section of this document.
2. In Object Explorer, connect to a SQL Server 2008 R2 instance and expand the Management node. Under the Management node, right-click the Data-tier Applications node, and then click **Deploy Data-tier Application**. The Deploy Data-tier Application Wizard appears.
3. Read the instructions in the Introduction page of the Deploy Data-tier Application Wizard, and then click **Next**.

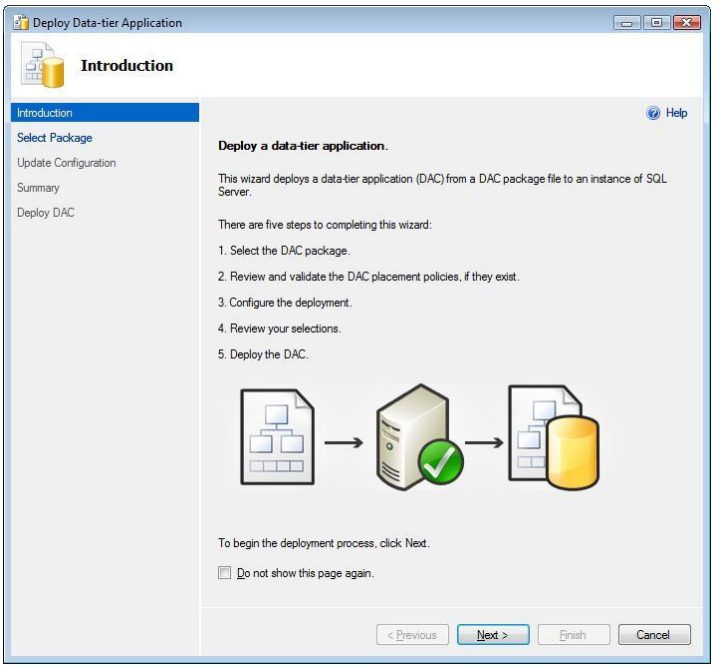

Figure 25: The Introduction page of the Deploy Data-tier Application Wizard

4. In the Select Package page, click **Browse**, and then select the DAC package file (with the .dacpac extension) that you want to deploy. Review the DAC details, and then click **Next**.

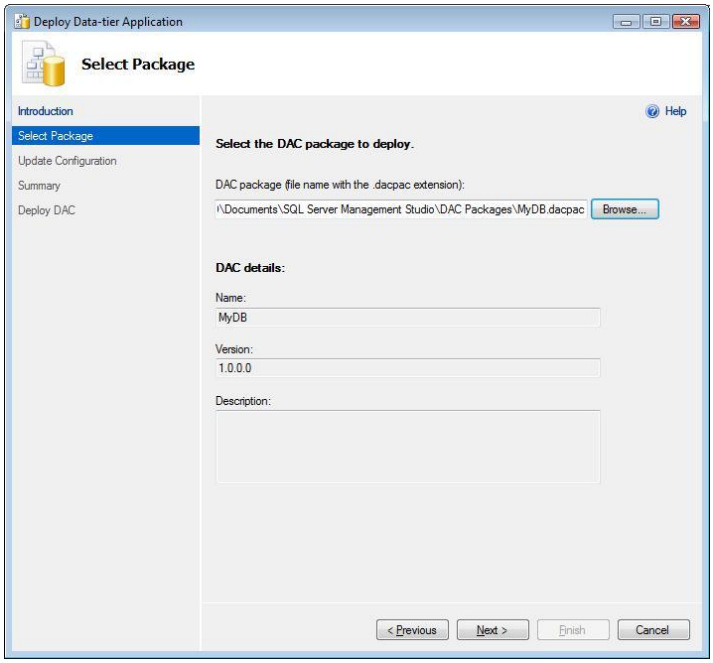

Figure 26: The Select Package page of the Deploy Data-tier Application Wizard

5. In the Update Configuration page, enter the parameters that will be used to deploy the DAC, namely the DAC name and the name of the database that will be created to host the database objects of the DAC. When you are finished, click **Next**.

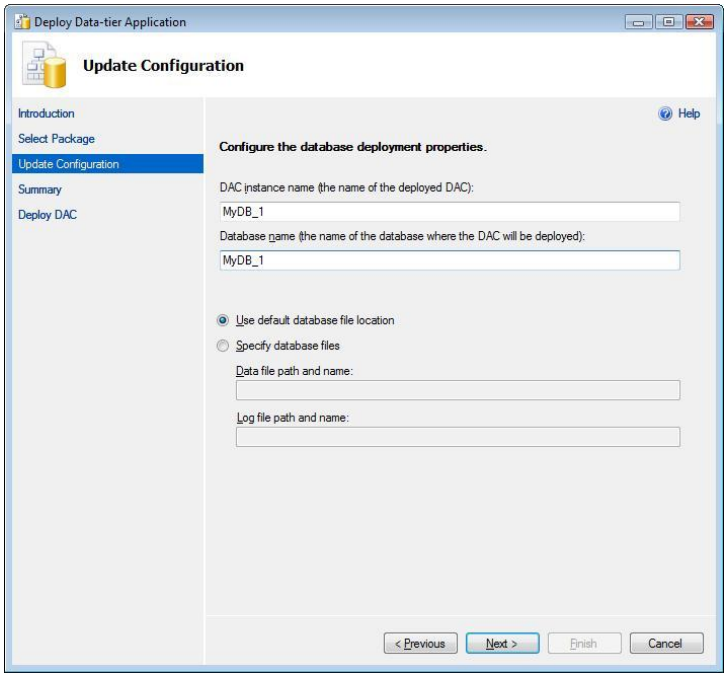

Figure 27: The Update Configuration page of the Deploy Data-tier Application Wizard

6. Verify the details listed in the Summary page of the wizard. To deploy the data-tier application you selected in step 4, click **Next**.

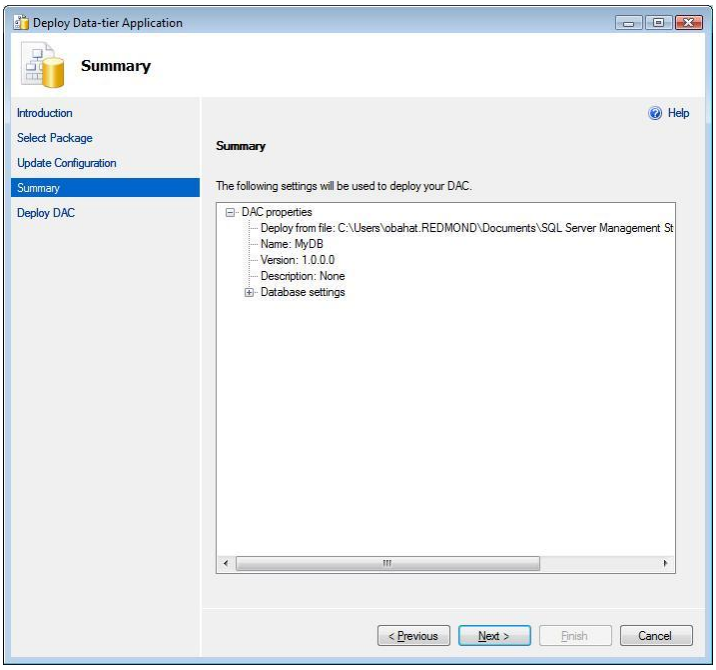

Figure 28: The Summary page of the Deploy Data-tier Application Wizard

7. The Deploy DAC page creates a new database, the database schema defined in the DAC, and logins that map to the database of the DAC. Review the steps taken by the wizard in this page, and then click **Finish** to complete the wizard.

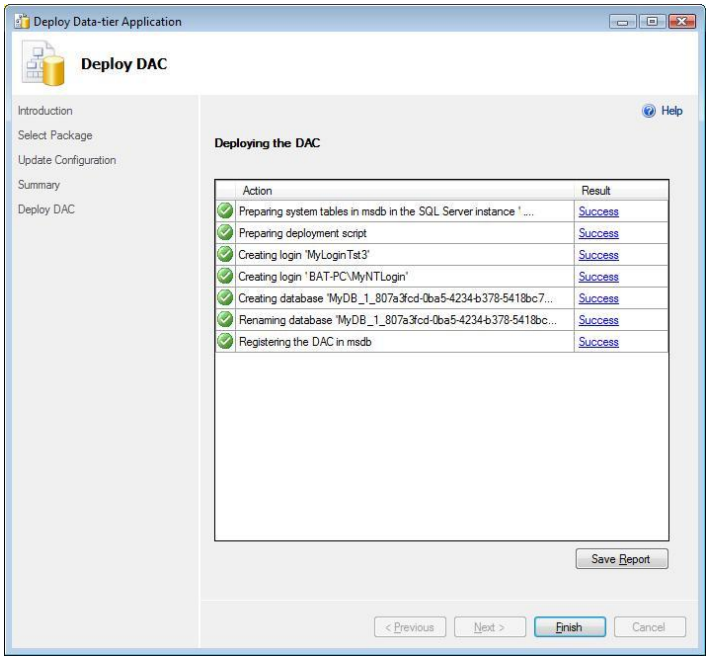

Figure 29: The Deploy DAC page of the Deploy Data-tier Application Wizard

8. In Object Explorer, expand the Databases node and locate the newly created database. The name of this database is the one you selected in step 5. Expand the database node and review the database objects that were created. Also, under the Management node, expand the Data-tier Applications, and then locate the DAC entry for the newly deployed DAC.

Conclusion

Familiar tools combined with new wizards help make setting up a SQL Server managed server group fast and easy. After managed server groups are set up, DBAs can easily assess capacity health and make decisions about consolidation to save money and better protect the health of their database environment. The introduction of the Data-tier Application introduces a single unit of deployment to accelerate consolidation and upgrade initiatives across the application lifecycle.

For more information:

http://www.microsoft.com/sqlserver/: SQL Server Web site

http://www.microsoft.com/sqlserver/2008/en/us/r2.aspx: SQL Server 2008 R2

http://technet.microsoft.com/en-us/sqlserver/: SQL Server TechCenter

http://msdn.microsoft.com/en-us/sqlserver/: SQL Server DevCenter

Did this paper help you? Please give us your feedback. Tell us on a scale of 1 (poor) to 5 (excellent), how would you rate this paper and why have you given it this rating? For example:

- Are you rating it high due to having good examples, excellent screen shots, clear writing, or another reason?
- Are you rating it low due to poor examples, fuzzy screen shots, or unclear writing?

This feedback will help us improve the quality of white papers we release.

Send feedback.