Database and Middleware Automation

Ultimate Edition

Software Version: 10.50

Workflows for SQL Server

Document Release Date: June 2016
Software Release Date: June 2016

Hewlett Packard Enterprise

Legal Notices

Warranty

The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from Hewlett Packard Enterprise required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notice

Trademark Notices

Adobe™ is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark of The Open Group.

This product includes an interface of the 'zlib' general purpose compression library, which is Copyright © 1995-2002 Jean-loup Gailly and Mark Adler.

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates or to verify that you are using the most recent edition of a document, go to: https://softwaresupport.hp.com/.

This site requires that you register for an HP Passport and to sign in. To register for an HP Passport ID, click **Register** on the HP Software Support site or click **Create an Account** on the HP Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HPE sales representative for details.

Support

Visit the HP Software Support site at: https://softwaresupport.hp.com.

This website provides contact information and details about the products, services, and support that HP Software offers.

HP Software online support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support website to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and to sign in. Many also require a support contract. To register for an HP Passport ID, click **Register** on the HP Support site or click **Create an Account** on the HP Passport login page.

To find more information about access levels, go to: https://softwaresupport.hp.com/web/softwaresupport/access-levels.

HP Software Solutions Now accesses the HPSW Solution and Integration Portal website. This site enables you to explore HP Product Solutions to meet your business needs, includes a full list of Integrations between HP Products, as well as a listing of ITIL Processes. The URL for this website is http://h20230.www2.hp.com/sc/solutions/index.jsp.

About this PDF Version of Online Help

This document is a PDF version of the online help. This PDF file is provided so you can easily print multiple topics from the help information or read the online help in PDF format. Because this content was originally created to be viewed as online help in a web browser, some topics may not be formatted properly. Some interactive topics may not be present in this PDF version. Those topics can be successfully printed from within the online help.

Contents

SQL Server 7
MS SQL - Compliance Audit v2 8
Prerequisites for this Workflow 9
How this Workflow Works 10
How to Run this Workflow 12
Sample Scenarios 16
Parameters for MS SQL - Compliance Audit v2 23
MS SQL - Install Patch 24
Prerequisites 24
Process Overview 25
Workflow: MS SQL - Install Patch 25
Solution pack 26
Parameters to expose 26
Input parameters 26
FAQs 28
How do I install the SQL Server patch on all instances on the server? 28
How do I install the SQL Server patch on multiple cluster nodes? 28
MS SQL - Install Cluster Patch 28
Prerequisites 29
Process Overview 29
Workflow: MS SQL - Install Cluster Patch 30
Solution pack 30
Parameters to expose 30
Input parameters 30
FAQs 32
How do I install the SQL Server patch on all instances on the server? 32
How do I install the SQL Server patch on multiple cluster nodes? 32
Refreshing Database 33
MS SQL - Backup Database 34
Prerequisites for this Workflow 35
How this Workflow Works 36

How to Run this Workflow 38
Sample Scenarios 40
Parameters for Backup MS SQL Database 42
MS SQL - Restore Database 45
Prerequisites for this Workflow 47
How this Workflow Works 48
How to Run this Workflow 52
Sample Scenarios 54
Parameters for Restore MS SQL Database 57
MS SQL - Backup and Restore Database 59
Prerequisites for this Workflow 61
How this Workflow Works 62
How to Run this Workflow 65
Sample Scenarios 67
Parameters for Backup and Restore MS SQL Database 71
DB Release for SQL Server v2 76
Prerequisites for this Workflow 79
How this Workflow Works 80
How to Run this Workflow 83
Sample Scenarios 88
Parameters for DB Release for SQL Server v2 92
MS SQL Drop Database 95
Prerequisites for this Workflow 96
How this Workflow Works 97
How to Run this Workflow 98
Parameters for MS SQL - Drop Database 100
MS SQL - Upgrade Standalone SQL Instance 100
Prerequisites for this Workflow 101
How this Workflow Works 102
How to Run this Workflow 104
Parameters for MS SQL - Upgrade Standalone SQL Instance 106
MS SQL Rollback Patch 108
Prerequisites for this Workflow 109
How this Workflow Works 110
How to Run this Workflow 112

Parameters for MS SQL Rollback Patch 114
MS SQL - Create AlwaysOn Availability Group 115
Prerequisites for this Workflow 116
How this Workflow Works 117
How to Run this Workflow 119
Parameters for MSSQL - Create AlwaysOn Availability Group 121
MS SQL - Install Clustered SQL Instance 122
Prerequisites 123
How this workflow works 124
How to run this workflow 126
Parameters for MS SQL - Install Clustered SQL Instance 130
MS SQL - Add Node to Cluster v2 136
Prerequisites 137
How this workflow works 138
How to run this workflow 139
Parameters for MS SQL - Add Node to Cluster 144
MS SQL - Create Database 147
Prerequisites 147
How this workflow works 148
How to run this workflow 149
Parameters for MS SQL - Create Database 153
Send documentation feedback 155

SQL Server

Workflow type	Workflow name
Compliance	"MS SQL - Compliance Audit v2" on the next page
Provisioning	"MS SQL - Install Clustered SQL Instance" on page 122
	"MS SQL - Add Node to Cluster v2" on page 136
	"MS SQL - Create Database" on page 147
	MS SQL - Install Standalone SQL Instance
	"MS SQL - Create Database" on page 147
	"MS SQL - Upgrade Standalone SQL Instance" on page 100
	"MS SQL - Create AlwaysOn Availability Group" on page 115
Patching	"MS SQL - Install Patch" on page 24
	"MS SQL - Install Cluster Patch" on page 28
	"MS SQL Rollback Patch" on page 108
Refreshing	"MS SQL - Backup Database" on page 34
	"MS SQL - Backup and Restore Database" on page 59
	"MS SQL - Restore Database" on page 45
Release Management	"DB Release for SQL Server v2" on page 76

MS SQL - Compliance Audit v2

The MS SQL - Compliance Audit workflow enables you to audit a Microsoft SQL Server instance for compliance with the following security benchmark requirements:

- Center for Internet Security (CIS) security configuration benchmarks
- Payment Card Industry (PCI) data security standard
- Sarbanes-Oxley (SOX) requirements

The workflow performs CIS Level 1 and Level 2 auditing for a SQL Server instance. The audit identifies compliance related problems with a SQL Server instance.

The workflow performs the checks included in the CIS benchmark and then maps those CIS checks to the benchmark type that you specify in the Compliance Type parameter. The audit summary email will match the Compliance Type that you specify.

To use this workflow in your environment, see the following information:

Topic	Information Included
"Prerequisites for this Workflow"	List of prerequisites that must be satisfied before you can run this workflow
"How this Workflow Works"	Information about what the workflow does, including validation checks performed, steps executed, and step descriptions
"How to Run this Workflow"	Instructions for running this workflow in your environment
"Sample Scenarios"	Examples of typical parameter values for this workflow
Parameters	List of input parameters for this workflow

Prerequisites for this Workflow

Be sure that the following prerequisites are satisfied before you run the MS SQL - Compliance Audit v2 workflow:

- The latest HPE DMA solution packs require the latest HPE DMA platform. To use the latest solution packs, update the HPE DMA platform. HPE DMA10.50 solution packs are supported on HPE DMA10.50 (and later).
- You have installed the Database Compliance solution pack.

The workflow must be able to:

- Execute `reg.exe` (Windows Server command-line registry tool), `wmic.exe` (Windows Management Instrumentation Command-line tool), and "`net`" Windows utilities on the target server. These utilities are included in the base Windows Server installations.
- Log in to the SQL Server instance using Windows-authenticated login credentials.
- Read system tables and execute system procedures upon connecting to the SQL Server instance.

For more information about prerequisites for Microsoft SQL Server, refer to the Microsoft SQL Server Documentation.

How this Workflow Works

This workflow performs the following actions:

- Prepares to run the workflow by gathering information about the target SQLServerInstance and validating parameter values.
- Audits the various configuration settings specified in the pertinent CIS, SOX, or PCI benchmark.
- Composes and sends an email containing the results of the audit.

> **Note:** The emails are sent through the mail server configured on the HPE DMA server. You can configure the mail server in the path below:
>
> DMA setup > Configuration > Outgoing Mail > Server.

Validation Checks Performed show

This workflow validates the following conditions:

1. Either `sqlcmd.exe` or `osql.exe` must be installed on the target machine.
2. Any Excluded Checks specified by the user refer to actual CIS, SOX, or PCI benchmark checks.
3. Any email addresses specified are valid addresses.
4. The workflow can create the temporary file that will store the compliance check results.

Steps Executed show

The MS SQL - Compliance Audit workflow includes the following steps. Each step must complete successfully before the next step can start. If a step fails, the workflow reports a failure and all subsequent steps are skipped.

Steps Used by Run MS SQL Compliance Audit

Workflow Step	Description
Gather Parameters for MS SQL Compliance	This step gathers two pieces of information: (1) the type of compliance audit to perform and (2) the list of compliance checks to exclude from the audit.
Gather Advanced Parameters for MS SQL Compliance	This step gathers the information that the workflow needs to create and deliver the compliance audit report via email. It also enables you to specify the name of the latest available SQL Server build and the Windows domain user.

Steps Used by Run MS SQL Compliance Audit, continued

Workflow Step	Description
Validate Compliance Parameters v2	This step validates the input parameters specified in the previous steps. It validates the list of excluded checks to ensure that all specified checks in the list correspond to actual Center for Internet Security (CIS) benchmark items. It also validates the email information to ensure that all specified email addresses are valid. The step then creates the path to the temporary file that will store the results of the current audit as the workflow is running. This file is deleted after the audit report is sent.
MS SQL Prepare SQL Server Compliance Check	This step determines whether workflow can perform the following actions on the target system: • Check database connectivity • Query the registry • Check the registry for SQL Server • Execute Windows Management Instrumentation (WMI) API calls • Execute the `net user /?` command If the workflow can perform all of these actions, it is capable of running the Center for Internet Security (CIS) Security Configuration Benchmark compliance tests.
MS SQL - Compliance Checks	This step executes all the compliance checks for MS SQL server.
Validate Post Compliance Checks	This step reads the temporary file that contains the results of the compliance audit and prints the audit results to the HPE DMA Console. It also creates (or updates) the compliance metadata fields for the target. If email addresses were specified, it also creates a report in HTML format that will be emailed to those addresses by a later step in the workflow.
Send Compliance Email	If email addresses are provided, this step sends the previously generated compliance audit report to the specified email addresses.
Delete File	This step deletes the specified file on the target server.

Note: For input parameter descriptions and defaults, see "Parameters for MS SQL - Compliance Audit v2" on page 23.

How to Run this Workflow

The following instructions show you how to customize and run the MS SQL - Compliance Audit v2 workflow in your environment.

The workflow provides default values for some parameters. These default values are usually sufficient for a "typical" installation. You can override the defaults by specifying parameter values in the deployment. You can also expose additional parameters in the workflow, if necessary, to accomplish more advanced scenarios. Any parameters not explicitly specified in the deployment will have the default values listed in "Parameters for MS SQL - Compliance Audit v2" on page 23.

> **Note:** Before following this procedure, review the "Prerequisites for this Workflow" on page 9, and ensure that all requirements are satisfied.

To use the MS SQL Compliance Audit v2 workflow:

1. Create a deployable copy of the workflow.
2. Determine the values that you will specify for the following parameters:

   Parameters Defined in this Step: Gather Parameters for SQL Server Compliance

Parameter Name	Default Value	Required	Description
Compliance Type	CIS	optional	Type of compliance report that will be generated by the workflow. Supported types are: CIS = Center for Internet Security (CIS) Security Configuration Benchmark PCI = Payment Card Industry (PCI) Data Security Standard SOX = Sarbanes-Oxley (SOX) sections 302.2, 302.4b, 302.4c, and 302.5 requirements
Excluded Checks	no default	optional	Comma-separated list of compliance checks to exclude from the audit. For example: 1.2, 2, 3.*, 5*, 6.1.2 **Note:** Make sure that the checks specified here

Parameters Defined in this Step: Gather Parameters for SQL Server Compliance , continued

Parameter Name	Default Value	Required	Description
			correspond with the compliance audit type (CIS, PCI, or SOX) that you are running.

Parameters Defined in this Step: Advanced Parameter for MS SQL Compliance

Parameter Name	Default Value	Required	Description
Email Addresses to Receive Report	no default	optional	Comma-separated list of email addresses for those individuals or groups who will receive a copy of the compliance audit report.
Instance Account	no default	optional	The Windows account that will perform the compliance audit.
Instance Password	no default	optional	The password for the Windows account that will perform the compliance audit.
Latest Build to Check for	no default	optional	The latest build of SQL server according to Microsoft. For example, build 4439 for SQL Server 2014 SP1.

> **Note:** This is the minimum set of parameters required to run this workflow. You may need to expose additional parameters depending on your objectives.
>
> See "Parameters for MS SQL - Compliance Audit v2" on page 23 for detailed descriptions of all input parameters for this workflow, including default values.

3. In the workflow editor, expose any additional parameters that you need. You will specify values for those parameters when you create the deployment or at runtime.
4. Save the changes to the workflow (click **Save** in the lower right corner).
5. Create a new deployment.

6. On the Parameters tab, specify values (or set the type to Runtime Value) for the required parameters listed in step 2 and any additional parameters that you have exposed. You do not need to specify values for those parameters whose default values are appropriate for your environment.
7. On the Targets tab, specify one or more targets for this deployment.
8. Save the deployment (click **Save** in the lower right corner).
9. Run the workflow using this deployment, specifying any runtime parameters.

To verify the results:

The workflow will complete and report SUCCESS on the Console if it has run successfully. If an error occurs during workflow execution, the error is logged, and the workflow terminates in the FAILURE state.

Information about each compliance check is displayed in the step output on the Console (and the History page) for each of the audit steps.

A summary of the compliance audit is also displayed in the step output for the Validate Post Compliance Checks step.

To view the reports:

A compliance audit summary in HTML format is emailed to all parties on the Email Addresses to Receive Report list.

After you run this workflow, you can generate two types of compliance reports on the Reports page:

- Database Compliance Report
- Database Compliance Detail Report

To access the Database Compliance reports:

1. Go to the Reports page.
2. At the bottom of the page,specify the following settings:

   For the Database Compliance Report:

   a. Select the Database Compliance report.

   b. Select the organization where your target resides.

   c. Because this report lists the latest compliance audit reports for all targets in the specified organization, you do not specify a Server, Database, or Time span.

   For the Database Compliance Detail Report:

   a. Select the Database Compliance Details report.

   b. Select the organization where your target resides.

   c. Specify the Server and Instance that you selected when you created your deployment.
3. Click **Run report**.

Sample Scenarios

This topic shows you how to use various parameters to achieve the following compliance audit scenarios in your environment using the "MS SQL - Compliance Audit v2" workflow.

Scenario 1: Perform a Partial CIS Compliance Audit and Email the Results show

In the scenario, the following checks are excluded from the audit:

- Section 7: Replication
- Section 9: Surface Area Configuration Tool

A summary report is sent to the three parties listed in the Email Addresses to Receive Report parameter.

Parameter Name	Example Value	Description
Compliance Type	CIS	Type of compliance report that will be generated by the workflow. Supported types are: CIS = Center for Internet Security (CIS) Security Configuration Benchmark PCI = Payment Card Industry (PCI) Data Security Standard SOX = Sarbanes-Oxley (SOX) sections 302.2, 302.4b, 302.4c, and 302.5 requirements
Excluded Compliance Checks	7.*,9.*	Comma-separated list of compliance checks to exclude from the audit. For example: 1.2, 2, 3.*, 5*, 6.1.2 **Note:** Make sure that the checks specified here correspond with the compliance audit type (CIS, PCI, or

Parameter Name	Example Value	Description
		SOX) that you are running.
Email Addresses to Receive Report	SQLDBAdminTeam@mycompany.com, SQLDBAdminMgr@mycompany.com, CustomerSupportTeam@mycompany.com	Comma-separated list of email addresses for those individuals or groups who will receive a copy of the compliance audit report.

Note: Some of these parameters are not exposed by default in the deployment.

Be sure that the default values for all remaining input parameters are appropriate for your environment (see "Parameters for MS SQL - Compliance Audit v2").

Scenario 2: Perform a Full PCI Compliance Audit and Email the Results show

A summary report is sent to the three parties listed in the Email Addresses to Receive Report parameter.

Parameter Name	Example Value	Description
Compliance Type	PCI	Type of compliance report that will be generated by the workflow. Supported types are: CIS = Center for Internet Security (CIS) Security Configuration Benchmark PCI = Payment Card Industry (PCI) Data Security Standard SOX = Sarbanes-Oxley (SOX) sections 302.2, 302.4b, 302.4c, and 302.5 requirements
Email Addresses to Receive Report	SQLDBAdminTeam@mycompany.com, SQLDBAdminMgr@mycompany.com, CustomerSupportTeam@mycompany.com	Comma-separated list of email addresses for those individuals or groups who

Parameter Name	Example Value	Description
		will receive a copy of the compliance audit report.

> **Note:** Some of these parameters are not exposed by default in the deployment.

Be sure that the default values for all remaining input parameters are appropriate for your environment (see "Parameters for MS SQL - Compliance Audit v2").

Scenario 3: Perform a Full SOX Compliance Audit, Email the Results, and Configure Windows Domain User Using Runtime Parameters show

A summary report is sent to the three parties listed in the Email Addresses to Receive Report parameter.

> **Note:** You may want to run this workflow against a MS SQL instance that can only be accessed by a Windows user with a temporary password. By using a runtime parameter for the password, you can ensure that the password used is always the latest.
>
> To specify the Windows domain user at the time you execute a deployment with runtime parameters, perform the following additional steps:
>
> 1. When you make a copy of the workflow, expand the appropriate step, and then set the Windows domain user parameters—Instance Account and Instance Password—to **- User selected -**.
> 2. When you create a deployment from the copy of the workflow, set the parameter types to **Runtime Value**.
> 3. When you execute the deployment, specify the Windows domain user account and password.

Parameter Name	Example Value	Description
Compliance Type	SOX	Type of compliance report that will be generated by the workflow. Supported types are: CIS = Center for Internet Security (CIS) Security Configuration Benchmark PCI = Payment Card Industry (PCI) Data Security Standard SOX = Sarbanes-Oxley (SOX) sections 302.2, 302.4b, 302.4c, and 302.5 requirements
Email Addresses to Receive Report	SQLDBAdminTeam@mycompany.com, SQLDBAdminMgr@mycompany.com, CustomerSupportTeam@mycompany.com	Comma-separated list of email addresses for those individuals or groups who will receive a copy of the

Parameter Name	Example Value	Description
		compliance audit report.
Instance Account	Domain\DomainUserAcct **Note:** Enter at runtime.	The Windows account that will perform the compliance audit.
Instance Password	DomainUserPswd **Note:** Enter at runtime.	The password for the Windows account that will perform the compliance audit.
Latest Build to Check for	5058	The latest build of Microsoft SQL Server 2005, according to Microsoft. Ensure that instance is at least patched up to indicated build level. Example value would be “5058” for SQL 2012’s SP2. If no value is given, the related Compliance check will be skipped.

Note: Some of these parameters are not exposed by default in the deployment.

Be sure that the default values for all remaining input parameters are appropriate for your environment (see "Parameters for MS SQL - Compliance Audit v2").

Scenario 4: Perform a Full CIS Compliance Audit and Display the Results on the HPE DMA Consoleshow

In the scenario, all scorable checks are performed, and the compliance audit report is displayed only on the HPE DMA Console. In this case, a summary report is not emailed. This scenario would be appropriate for initial testing.

It is not necessary to specify any input parameters in this scenario unless the SQL Server inventory file is located in a non-standard directory.

Parameter Name	Example Value	Description
Compliance Type	CIS	Type of compliance report that will be generated by the workflow. Supported types are: CIS = Center for Internet Security (CIS) Security Configuration Benchmark PCI = Payment Card Industry (PCI) Data Security Standard SOX = Sarbanes-Oxley (SOX) sections 302.2, 302.4b, 302.4c, and 302.5 requirements

Be sure that the default values for all remaining input parameters are appropriate for your environment (see "Parameters for MS SQL - Compliance Audit v2").

Parameters for MS SQL - Compliance Audit v2

The following tables describe the required and optional input parameters for this workflow. Some of these parameters are not initially visible in a deployment. For most parameters, if you do not specify a value for a parameter, a default value is assigned.

> Only those parameters that are configurable in a standard deployment are listed here. Input parameters that must be mapped to output parameters of previous steps are not listed.

Parameters Defined in this Step: Gather Parameters for MS SQL Compliance

Parameter Name	Default Value	Required	Description
Compliance Type	CIS	optional	Type of compliance report that will be generated by the workflow. Supported types are: CIS = Center for Internet Security (CIS) Security Configuration Benchmark PCI = Payment Card Industry (PCI) Data Security Standard SOX = Sarbanes-Oxley (SOX) sections 302.2, 302.4b, 302.4c, and 302.5 requirements
Excluded Checks	no default	optional	Comma-separated list of compliance checks to exclude from the audit. For example: 1.2, 2, 3.*, 5*, 6.1.2 **Note:** Make sure that the checks specified here correspond with the compliance audit type (CIS, PCI, or SOX) that you are running.

Parameters Defined in this Step: Gather Advanced Parameters for MS SQL Compliance

Parameter Name	Default Value	Required	Description
Email Addresses to Receive Report	no default	optional	Comma-separated list of email addresses for those individuals or groups who will receive a copy of the compliance audit report.
Instance Account	no default	optional	The Windows account that will perform the compliance audit.
Instance Password	no default	optional	The password for the Windows account that will perform the compliance audit.
Latest Build	no default	optional	The latest build of Microsoft SQL Server 2005,

Parameters Defined in this Step: Gather Advanced Parameters for MS SQL Compliance, continued

Parameter Name	Default Value	Required	Description
to Check for			according to Microsoft. Ensure that instance is at least patched up to indicated build level. Example value would be “5058” for SQL 2012’s SP2. If no value is given, the related Compliance check will be skipped.

MS SQL - Install Patch

This section describes how to use Database and Middleware Automation (HPE DMA) to create a repeatable, standardized method to quickly and accurately install Microsoft Microsoft SQL Server patches on SQL Server installations across an enterprise to reach patch currency standards.

> **Tip:** To patch more complex SQL Server clustered environments, see *Achieve Patch Currency for Microsoft SQL Server Clustered Environments Using HPE DMA*, available at: https://softwaresupport.hp.com/

Prerequisites

Before performing the procedures in this section, your environment must meet the following minimum requirements:

- A server running Windows 2008, 2008 R2, or 2012.
- A SQL Server instance—version 2005, 2008, 2008R2, or 2012—is provisioned and ready to be patched.
- Patch media:

  The SQL Server patch file from Microsoft.

  Patch installation media must be available locally or available for download from the software repository.

- Storage: A staging directory with 1 gigabyte available.
- Licenses for SQL Server and HPE DMA.

Process Overview

Installing a SQL Server patch to a Microsoft SQL Server installation with HPE DMA is a simple, one-step process. All required checks and steps have been implemented in a single HPE DMA workflow.

Use the following HPE DMA workflow to standardize the process of installing a SQL Server patch:

HPE DMA can install any of the following types of SQL Server patches:

- Hot Fixes
- Cumulative Updates
- Service Packs

> **Note:** This workflow patches a single SQL Server instance unless you use the use the advanced parameter Patch All Instances on Server. The advanced parameter is demonstrated in this section.

> **Tip:** To patch multiple SQL Server cluster nodes, run MS SQL - Install Patch once for each node, or for an easier process, use the MS SQL - Install Cluster Patch workflow that is described in *Achieve Patch Currency for Microsoft SQL Server Clustered Environments Using HPE DMA*, available at: https://softwaresupport.hp.com/

Workflow: MS SQL - Install Patch

This section provides detailed information required to run the MS SQL - Install Patch workflow.

> **Tip:** To patch multiple SQL Server cluster nodes, run MS SQL - Install Patch once for each.

Solution pack

This workflow requires the Database Patching Solution Pack.

Parameters to expose

If you want to patch all SQL Server instances, in the workflow's MS SQL - Advanced Parameters - Install Patch step, expose the Patch All Instances on Server parameter.[1]

Input parameters

When you deploy the MS SQL - Install Patch workflow, specify input parameter values for the following steps.

> Bold text in the following tables indicates that you must specify a value for the parameter.

Step: MS SQL - Parameters - Install Patch

Parameter	Description	Example Value
Download From	Required: The name of the SQL	`SQL12_SP1.exe`

[1] This parameter is hidden by default and must be exposed when you make a copy of the workflow.

Step: MS SQL - Parameters - Install Patch, continued

Parameter	Description	Example Value
Software Directory	Server patch file obtained from Microsoft.[1] **Note:** This must be an EXE file. If you obtain a ZIP file from Microsoft, unzip it to retrieve the EXE file.	
Download Target Destination	Required: The local directory where the SQL Server patch file is stored: If patch file is in the software repository: Location where Download From Software Directory will be downloaded If patch file is on the target: Location where the Microsoft SQL Server patch file already exists Upon a successful workflow completion, all downloaded files are cleaned up.	`C:\temp`
Web Service Password	Required: Password for the HPE DMA Discovery web service API.	●●●
Web Service User	Required: User who is capable of modifying the managed environment by using the HPE DMA Discovery web service API.	dmawebuser

Step: MS SQL - Advanced Parameters - Install Patch

Parameter	Description	Example Value
Patch All Instances on Server	Optional: Flag to determine whether all SQL Server instances on the server will be patched. Valid values: Yes or No. Default: No.	Yes

[1] If the file is not found on the target server(s), it will be downloaded from the software repository.

Step: MS SQL Kill Processes

Parameter	Description	Example Value
Instance Account	Optional: The Windows account that will terminate the SQL Server processes.	

FAQs

How do I install the SQL Server patch on all instances on the server?

To install the SQL Server patch on all instances on the server, set the Patch All Instances on Server parameter to Yes before you execute the deployment:

Workflow: MS SQL - Install Patch

Step: MS SQL - Advanced Parameters - Install Patch

Parameter: Patch All Instances on Server[1]

How do I install the SQL Server patch on multiple cluster nodes?

To install the SQL Server patch on multiple cluster nodes, run the MS SQL - Install Patch workflow once on each cluster node.

MS SQL - Install Cluster Patch

This section describes how to use Database and Middleware Automation (HPE DMA) to create a repeatable, standardized method to quickly and accurately install Microsoft SQL Server patches on SQL Server clustered installations across an enterprise to reach patch currency standards.

[1]This parameter is hidden by default and must be exposed when you make a copy of the workflow.

> **Tip:** To patch SQL Server standalone environments, see *Achieve Patch Currency for Microsoft SQL Server Environments Using HPE DMA*, available at: softwaresupport.hp.com

Prerequisites

Before performing the procedures in this section, your environment must meet the following minimum requirements:

- A server running Windows 2008, 2008 R2, or 2012.
- A SQL Server clustered instance—version 2008, 2008 R2, or 2012—is provisioned and ready to be patched.
- Patch media:

  The SQL Server patch file from Microsoft.

  Patch installation media must be available locally or available for download from the software repository.
- Storage: A staging directory with 1 gigabyte available.
- Licenses for SQL Server and HPE DMA.

Process Overview

Installing a SQL Server patch to a Microsoft SQL Server clustered installation with HPE DMA is a simple, one-step process. All required checks and steps have been implemented in a single HPE DMA workflow.

HPE DMA can install any of the following types of SQL Server patches:

- Hot Fixes
- Cumulative Updates
- Service Packs

> **Note:** To execute the workflow, only one of the nodes in the SQL Server cluster needs to be a target for the deployment. The workflow discovers all cluster members and patches each one.

The following section provides detailed information required to run the workflow.

Workflow: MS SQL - Install Cluster Patch

This section provides detailed information required to run the MS SQL - Install Cluster Patch workflow.

Solution pack

This workflow requires the Database Patching Solution Pack.

Parameters to expose

None

Input parameters

When you deploy the MS SQL - Install Cluster Patch workflow, specify input parameter values for the following steps.

Step: MS SQL - Parameters - Install Patch

Parameter	Description	Example Value
Download From	Required: The name of the SQL	`SQL12_SP1.exe`

Step: MS SQL - Parameters - Install Patch, continued

Parameter	Description	Example Value
Software Directory	server patch file obtained from Microsoft.[1] **Note:** This must be an EXE file. If you obtain a ZIP file from Microsoft, unzip it to retrieve the EXE file.	
Download Target Destination	Required: The local directory where the SQL server patch file is stored: If patch file is in the software repository: Location where Download From Software Directory will be downloaded If patch file is on the target: Location where the Microsoft SQL server patch file already exists Upon a successful workflow completion, all downloaded files are cleaned up.	`C:\temp`
Web Service Password	Required: Password for the HPE DMA Discovery web service API.	●●●

Note: The step Run Subflow - MS SQL - Install Patch runs first to patch all passive nodes.

Step: Run Subflow - MS SQL - Install Patch

Parameter	Description	Example Value
Server Parallel Execution	Optional: Flag to determine whether the workflow is to execute in parallel. Set to False if you would like the workflow to execute serially. Default is True.	True

Note: The step Run Subflow - MS SQL - Install Patch runs again to patch the active node.

[1] If the file is not found on the target server(s), it will be downloaded from the software repository.

Step: Run Subflow - MS SQL - Install Patch

Parameter	Description	Example Value
Server Parallel Execution	Optional: Flag to determine whether the workflow is to execute in parallel. Set to False if you would like the workflow to execute serially. Default is True.	True

FAQs

How do I install the SQL Server patch on all instances on the server?

To install the SQL Server patch on all instances on the server, set the Patch All Instances on Server parameter to Yes before you execute the deployment:

Workflow: MS SQL - Install Patch

Step: MS SQL - Advanced Parameters - Install Patch

Parameter: Patch All Instances on Server[1]

How do I install the SQL Server patch on multiple cluster nodes?

To install the SQL Server patch on multiple cluster nodes, run the MS SQL - Install Patch workflow once on each cluster node.

[1]This parameter is hidden by default and must be exposed when you make a copy of the workflow.

Refreshing Database

This section describes the SQL Server workflows included in the HPE Database and Middleware Automation (HPE DMA) Database Refresh solution pack.

Database refresh involves copying the contents of one database into a database in the same or another SQL Server instance. This is useful, for example, if you want to move a database from a traditional IT infrastructure to a private cloud. It is also useful if you want to duplicate production data in a test environment for application development or troubleshooting purposes.

The workflows in this solution pack enable you to automate and simplify the following operations:

- Extracting the contents of a database into a backup file
- Restoring a database from an existing backup file
- Extracting the contents of one database and loading them into another database using a single **bridged execution** workflow that performs both steps

The workflows perform extensive validation checks prior to and immediately after the database backup and restore operations to ensure that the refresh is successful.

After a refresh is completed, the restore workflows can re-create any existing database users and roles.

The workflows can create or utilize a database backup file that is compressed, encrypted, or both.

MS SQL - Backup Database

This workflow enables you to backup a SQL Server database into file (the backup file) that is stored either locally or on a network share.

You can specify various options for the backup operation, including whether the backup file is compressed or encrypted with a password.

The workflow performs extensive validation checks prior to and immediately after the backup operation to ensure that the backup file is valid. The workflow will perform an additional integrity check on the backup file if you set the Perform Integrity Check parameter to YES.

To use this workflow in your environment, see the following information:

Topic	Information Included
"Prerequisites for this Workflow"	List of prerequisites that must be satisfied before you can run this workflow
"How this Workflow Works"	Information about what the workflow does, including validation checks performed, steps executed, and a high-level process flow
"How to Run this Workflow"	Instructions for running this workflow in your environment
"Sample Scenarios"	Examples of typical parameter values for this workflow
"Parameters for Backup MS SQL Database"	List of input parameters for this workflow

The process of deploying and running this workflow is the same for all scenarios, but the parameters required will differ depending on the specific scenario that you are implementing.

The workflow provides default values for most parameters. These default values are usually sufficient for a "typical" database backup. You can override the defaults by specifying parameter values in the deployment. You can also expose additional parameters in the workflow, if necessary, to accomplish more advanced scenarios.

Any parameters not explicitly specified in the deployment will have the default values listed in "Parameters for Backup MS SQL Database" on page 42 .

Prerequisites for this Workflow

The following prerequisites must be satisfied before you can run the "MS SQL - Backup Database" workflow:

1. The service login for the SQL Server service must have read and write permissions on the backup path.
2. The server management agent must have login access to the SQL Server instance in which the target database resides. It must also have permission to perform database consistency check (DBCC) commands on the target database.
3. There must be sufficient space available on the target data and log disks. The workflow checks for this, and will fail if sufficient space is not available.

Additional Considerations

For information about prerequisites for SQL Server, refer to the SQL Server Product Documentation.

How this Workflow Works

This topic contains the following information about the "MS SQL - Backup Database" workflow:

Validation Checks Performed

The workflow checks the following things prior to dumping the database. If any of these checks fails, the workflow fails.

1. All required parameters have values. If any required parameter does not have a value—either a value that you specify or a default value—the workflow fails in the Run MS SQL Pre-Backup Validation step.
2. The Target Backup Path is accessible, either locally or on a network share.

   If the Target Backup Path is on a network share, the Windows Share User has read and write access the share.
3. The target database exists, and the workflow can connect to it.
4. Adequate disk space is available to store the database backup file.
5. If the Target Backup Path does not currently exist, it will be created prior to creating the backup file.

Steps Executed

The "MS SQL - Backup Database" workflow includes the following steps. Each step must complete successfully before the next step can start. If a step fails, the workflow reports a failure, and all subsequent steps are skipped.

Process Flow

This workflow performs the following tasks:

1. Performs the preliminary validation checks described above.
2. Performs the database backup operation to create the backup file.
3. Performs post-backup validation checks to ensure that all required parameters had valid values.
4. If Perform Integrity Check was set to YES, performs an integrity check on the backup file.

Tips and Best Practices

It is good practice to run basic database consistency checks (DBCCs) on the source database before running this workflow to ensure that there are no internal errors in the database.

If you find errors in the source database, be sure to fix them before running this workflow. The workflow does not have the ability to diagnose or remediate problems in the database prior to performing the database backup.

How to Run this Workflow

This topic explains how to customize and run the "MS SQL - Backup Database" workflow in your environment.

> **Note:** Prior to running this workflow, review the "Prerequisites for this Workflow", and ensure that all requirements are satisfied.

To customize and run the Backup MS SQL Database workflow:

1. Create a deployable copy of the workflow.
2. Determine the values that you will specify for the following parameter. This is the minimum set of parameters required to run this workflow.

Parameter Name	Default Value	Description
Target Backup Path	no default	Where the database backup file will be stored, either locally or on a network share. You can specify both the path and file name, or you can specify only the path. ◦ If you specify a file name, it must end in `.bak`. ◦ If you do not specify a file name, the backup file name will have the following form: `<dataBaseName>_<dateTime>.bak` where `<dataBaseName>` represents the name of the target database specified when the workflow runs, and `<dateTime>` is the date and time when the Run MS SQL Pre-Backup Validation step is executed. If the file will be stored on a network share, the Windows Share User must have read and write access to that share.

3. See "Parameters for Backup MS SQL Database" on page 42 for detailed descriptions of all input parameters for this workflow, including default values.In the workflow editor, expose any additional parameters that you need. You will specify values for those parameters when you create the deployment or at runtime.
4. Save the changes to the workflow (click **Save** in the lower right corner).
5. Create a new deployment.
6. On the Parameters tab, specify values (or set the type to Runtime Value) for the required parameters listed in step 2 and any additional parameters that you have exposed. You do not need to specify values for those parameters whose default values are appropriate for your environment.

7. On the Targets tab, specify one or more targets for this deployment.
8. Save the deployment (click **Save** in the lower right corner).
9. Run the workflow using this deployment, specifying any runtime parameters.

The workflow will complete and report “Success” on the Console if it has run successfully. If an invalid parameter value is specified, an error is logged, and the workflow terminates in the “Failure” state.

Sample Scenarios

This topic shows you how to use various parameters to achieve the following database backup scenarios in your environment using the "MS SQL - Backup Database" workflow:

Scenario 1: Create a Backup File that is Not Encrypted or Compressed

This is the simplest SQL Server database backup scenario. In this example, the backup file is stored on a network share.

Step Name	Parameter Name	Example Value
Gather Parameters for MS SQL Database Backup	Target Backup Path	`\\WIN-DOMAIN-CTRL\Backups`
Gather Advanced Parameters for MS SQL Database Backup	Windows Share Password	WinSharePwd To avoid having to re-enter passwords whenever they change, you can create a policy to provide them to the workflow.
	Windows Share User	WIN\Administrator

Be sure that the default values for all remaining parameters are appropriate for your environment (see "Parameters for Backup MS SQL Database" on page 42).

Scenario 2: Create a Backup File that is Encrypted and Compressed

This scenario requires you to specify the encryption password and compression option for the database backup file. In this example, the backup file is stored in locally on the server that hosts the target database.

Step Name	Parameter Name	Example Value
Gather Parameters for MS SQL Database Backup	Target Backup Path	`c:\Backups\mytestdb_03122012.bak`
Gather Advanced Parameters for MS SQL Database Backup	Backup Encryption Password	EncryptMyBackup
	Compress Backup File	YES

Be sure that the default values for all remaining parameters are appropriate for your environment (see "Parameters for Backup MS SQL Database" on page 42).

Scenario 3: Create a Backup File, Perform an Integrity Check, and Configure Windows Domain User Using Runtime Parameters

This scenario runs an integrity check on the backup file after the backup is performed. In this example, the backup file is stored locally on the server that hosts the target database.

> **Note:** You may want to run this workflow against a MS SQL instance that can only be accessed by a Windows user with a temporary password. By using a runtime parameter for the password, you can ensure that the password used is always the latest.
>
> To specify the Windows domain user at the time you execute a deployment with runtime parameters, perform the following additional steps:
>
> 1. When you make a copy of the workflow, expand the appropriate step, and then set the Windows domain user parameters—Instance Account and Instance Password—to **- User selected -**.
> 2. When you create a deployment from the copy of the workflow, set the parameter types to **Runtime Value**.
> 3. When you execute the deployment, specify the Windows domain user account and password.

Step Name	Parameter Name	Example Value
Gather Parameters for MS SQL Database Backup	Target Backup Path	`c:\Backups\mytestdb_ 03122012.bak`
Gather Advanced Parameters for MS SQL Database Backup	Perform Integrity Check	YES
	Instance Account	Domain\DomainUserAcct **Note:** Enter at runtime.
	Instance Password	DomainUserPswd **Note:** Enter at runtime.

Be sure that the default values for all remaining parameters are appropriate for your environment (see "Parameters for Backup MS SQL Database" on the next page).

Parameters for Backup MS SQL Database

The following tables describe the required and optional input parameters for this workflow. Most of these parameters are not initially visible in a deployment. For most parameters, if you do not specify a value for a parameter, a default value is assigned

Parameters Defined in this Step: Gather Parameters for MS SQL Database Backup

Parameter Name	Default Value	Required	Description
Target Backup Path	no default	required	Where the database backup file will be stored, either locally or on a network share. You can specify both the path and file name, or you can specify only the path. • If you specify a file name, it must end in `.bak`. • If you do not specify a file name, the backup file name will have the following form: *<dataBaseName>*_*<dateTime>*.bak where *<dataBaseName>* represents the name of the target database specified when the workflow runs, and *<dateTime>* is the date and time when the Run MS SQL Pre-Backup Validation step is executed. If the file will be stored on a network share, the Windows Share User must have read and write access to that share.

Additional Parameters Defined in this Step: Gather Advanced Parameters for MS SQL Database Backup

Parameter Name	Default Value	Required	Description
Backup Description	no default	optional	Text that describes this backup (up to 255 characters).
Backup Encryption Password	no default	optional	To encrypt the backup file with a password, specify the password in this parameter. If you perform the backup using a password, you must also specify that password when you perform the restore.
Backup Name	no default	optional	The name of this backup (up to 128 characters).

Additional Parameters Defined in this Step: Gather Advanced Parameters for MS SQL Database Backup, continued

Parameter Name	Default Value	Required	Description
Compress Backup File	NO	optional	If you specify YES, the backup file will be compressed. Valid values: YES or NO. Compression is supported on SQL Server 2008 Enterprise and later. If you are running SQL 2005, and this parameter is set to YES, the workflow will ignore this value and continue without compression.
Expiration Date	no default	optional	Date and time when the backup file expires and the backup data is no longer considered relevant.After this date and time, SQL Server is not prevented from overwriting this backup file. The Expiration Date must be specified in a format compatible with the configured system datetime format. If both the Retention Days and the Expiration Date parameters are specified, the Retention Days parameter takes precedence.
Instance Account	no default	optional	The Windows account that will perform the backup operation.
Instance Password	no default	optional	The password for the Windows account that will perform the backup operation.

Additional Parameters Defined in this Step: Gather Advanced Parameters for MS SQL Database Backup, continued

Parameter Name	Default Value	Required	Description
Perform Integrity Check	NO	optional	If you specify YES, the workflow will perform an integrity check on the database backup file. Valid values: YES or NO.
Retention Days	no default	optional	Number of days after which the backup data is no longer considered relevant.After this number of days, SQL Server is not prevented from overwriting this backup file. If both the Retention Days and the Expiration Date parameters are specified, the Retention Days parameter takes precedence.
Windows Share Password	no default	optional	Password for the user specified in Windows Share User.
Windows Share User	no default	optional	Windows user who can access the specified Windows network share and who will own (and write) the backup file.

MS SQL - Restore Database

This workflow enables you to restore a SQL Server database from a previously created database backup file that is stored locally, on a network share, or in the software repository.

If the database does not exist in the target instance, the workflow will create it. If the database already exists, you can specify whether you want the workflow to overwrite its contents. You can also specify whether existing database users should be re-created after the restore operation—in which case, any users included in the backup file are ignored.

> **Note:** The parameters required to activate these options are hidden by default.

This workflow also provides a "simulation mode" where the Run MS SQL Pre-Restore Validation step is executed, but the restore is not performed. This is useful for testing or troubleshooting your parameter values.

The workflow performs extensive validation checks prior to and immediately after the restore operation to ensure that both the backup file and the restored database are valid.

The process of deploying and running this workflow is the same for all scenarios, but the parameters required will differ depending on the specific scenario that you are implementing.

The workflow provides default values for most parameters. These default values are usually sufficient for a "typical" database refresh. You can override the defaults by specifying parameter values in the deployment. You can also expose additional parameters in the workflow, if necessary, to accomplish more advanced scenarios.

Topic	Information Included
"Prerequisites for this Workflow"	List of prerequisites that must be satisfied before you can run this workflow
"How this Workflow Works"	Information about what the workflow does, including validation checks performed, steps executed, and a high-level process flow
"How to Run this Workflow"	Instructions for running this workflow in your environment
"Sample Scenarios"	Examples of typical parameter values for this workflow
"Parameters for Restore MS SQL Database"	List of input parameters for this workflow

The process of deploying and running this workflow is the same for all scenarios, but the parameters required will differ depending on the specific scenario that you are implementing.

The workflow provides default values for most parameters. These default values are usually sufficient for a "typical" database restore. You can override the defaults by specifying parameter values in the deployment. You can also expose additional parameters in the workflow, if necessary, to accomplish more advanced scenarios.

Any parameters not explicitly specified in the deployment will have the default values listed in "Parameters for Restore MS SQL Database" on page 57 .

Prerequisites for this Workflow

The following prerequisites must be satisfied before you can run the "MS SQL - Restore Database" workflow:

1. The service login for the SQL Server service must have read and write permissions on the backup file.
2. The server management agent must have login access to the target SQL Server instance. It must also have permission to create a new database and perform database consistency check (DBCC) commands on the restored database.
3. There must be sufficient space available on the target data and log disks. The workflow checks for this, and will fail if sufficient space is not available.

Additional Considerations

For information about prerequisites for SQL Server, refer to the SQL Server Product Documentation.

How this Workflow Works

This topic contains the following information about the "MS SQL - Restore Database" workflow:

Validation Checks Performed

The workflow checks the following things prior to dumping the database. If any of these checks fails, the workflow fails.

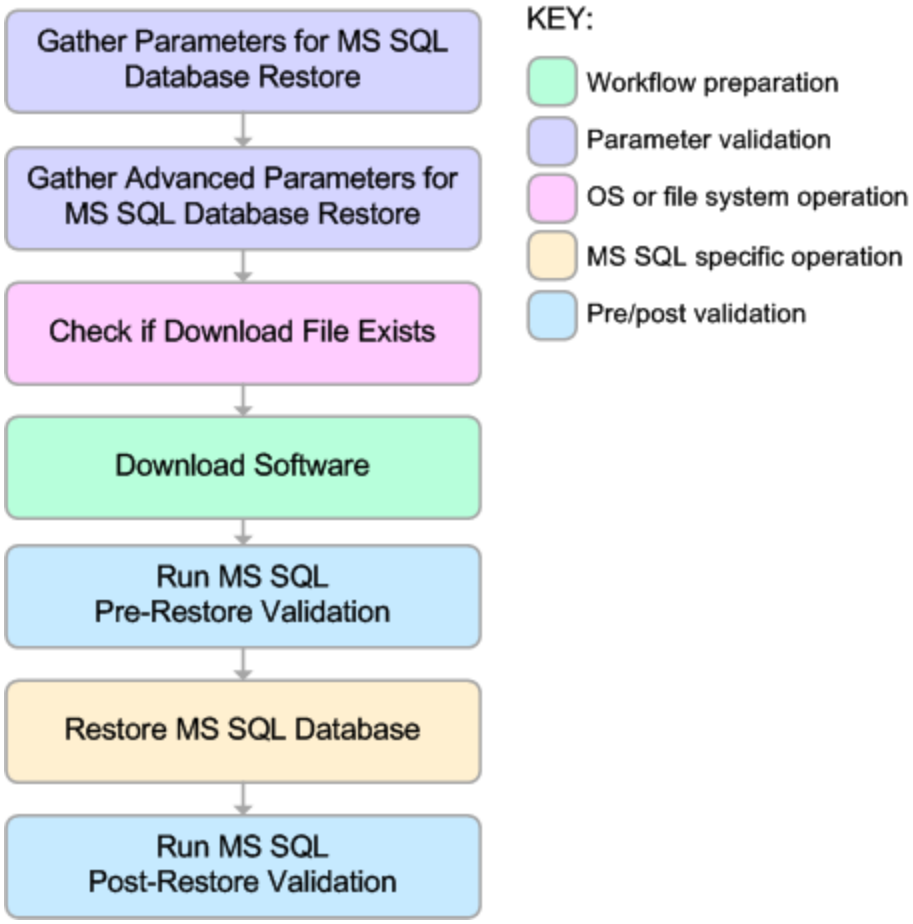
1. All required parameters have values. If any required parameter does not have a value—either a value that you specify or a default value—the workflow fails in the Run MS SQL Pre-Restore Validation step.
2. The specified backup file either exists in the Download Target Destination directory or can be downloaded from the software repository.
3. The backup file is compatible with the target instance.
4. If the Custom Database Name parameter is specified, this database name complies with SQL Server database naming conventions.
5. The Download Target Destination is accessible, either locally or on a network share.

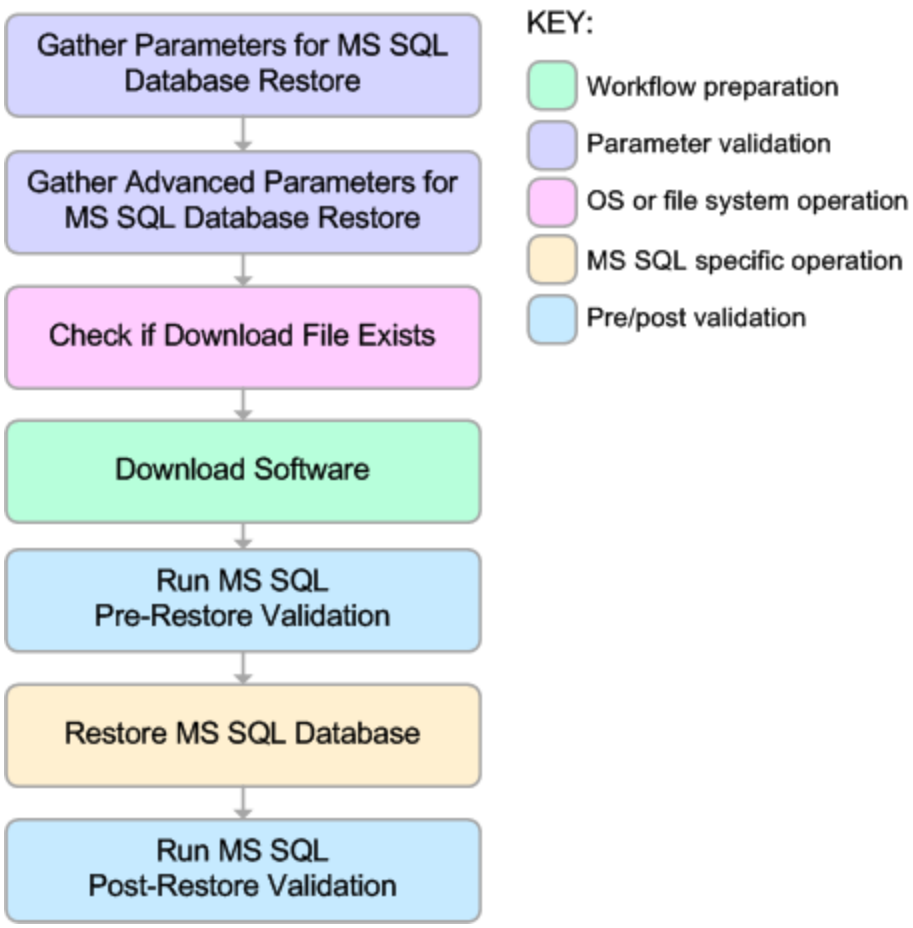
   If the Download Target Destination is on a network share, the Windows Share User has read and write access the to share.
6. The target instance exists, and the workflow can connect to it.
7. Adequate disk space is available to restore the data and log files.
8. If custom paths are specified for the data or log files, the Run MS SQL Pre-Restore Validation step checks that they exist (and creates them if they don't), and ensures that the quantity of paths specified match the quantity of files in the backup file.

Steps Executed

The "MS SQL - Restore Database" workflow includes the following steps. Each step must complete successfully before the next step can start. If a step fails, the workflow reports a failure, and all subsequent steps are skipped.

Click each box in the diagram to view additional information about that step in a new window.





Process Flow

This workflow performs the following tasks:

1. Performs the preliminary validation checks described above.
2. If Preserve Users and Roles was set to YES, creates the Roles Creation Script and the Users Creation Script script.
3. If not in simulation mode, performs the database restore operation to load the contents of the backup file.
4. Performs post-restore validation checks to ensure that the restored database is sound.
5. If Preserve Users and Roles was set to YES, re-creates any existing database users and roles.
6. If Reindex Restored Database was set to YES, re-indexes the database.

Tips and Best Practices

It is good practice to run basic database consistency checks (DBCCs) on the source database before you create the database backup to ensure that there are no internal errors in the database.

If you find errors in the source database, be sure to fix them before you create the database backup. This workflow does not have the ability to diagnose or remediate problems in the database prior to performing the database backup.

How to Run this Workflow

This topic explains how to customize and run the "MS SQL - Restore Database" workflow in your environment.

> **Note:** Prior to running this workflow, review the "Prerequisites for this Workflow", and ensure that all requirements are satisfied.

To customize and run the Restore MS SQL Database workflow:

1. Create a deployable copy of the workflow (see "Create a Deployable Workflow" in HPE DMA Quick Start Tutorial).
2. Determine the values that you will specify for the following parameters. This is the minimum set of parameters required to run this workflow.

Parameter Name	Default Value	Description
Database Backup File	no default	Path where the database backup file is (or will be) stored, either locally or on a network share. If the file already exists locally or on a network share, specify the file name in this parameter and the path in the Download Target Destination parameter. If the file does not yet exist locally or on a network share, it will be downloaded into this location from the software repository. If the file is (or will be) stored on a network share, the Windows Share User must have read and write access to that share. **Note:** Windows Share User and Windows Share Password are not exposed by default.
Download Target Destination	no default	The directory where the database backup file will be stored. If the database backup file does not yet exist in this directory, it will be downloaded from the software repository and stored in this directory.

   See "Parameters for Restore MS SQL Database" on page 57 for detailed descriptions of all input parameters for this workflow, including default values.
3. In the workflow editor, expose any additional parameters that you need. You will specify values for those parameters when you create the deployment or at runtime.
4. Save the changes to the workflow (click **Save** in the lower right corner).

5. Create a new deployment.
6. On the Parameters tab, specify values (or set the type to Runtime Value) for the required parameters listed in step 2 and any additional parameters that you have exposed. You do not need to specify values for those parameters whose default values are appropriate for your environment.
7. On the Targets tab, specify one or more targets for this deployment.
8. Save the deployment (click **Save** in the lower right corner).
9. Run the workflow using this deployment, specifying any runtime parameters.

The workflow will complete and report “Success” on the Console if it has run successfully. If an invalid parameter value is specified, an error is logged, and the workflow terminates in the “Failure” state.

Sample Scenarios

This topic shows you how to use various parameters to achieve the following database backup scenarios in your environment using the "MS SQL - Restore Database" workflow:

Scenario 1: Restore from a Backup File that is Not Encrypted or Compressed

This is the simplest SQL Server database restore scenario. In this example, the backup file has been stored on a network share (or will be downloaded from the software repository and stored on the share).

Note that the Windows Share User and Windows Share Password are specified in this scenario. This is not required, but it facilitates the disk space check on the network path. If you do not specify this parameter, this check is skipped.

Step Name	Parameter Name	Example Value
Gather Parameters for MS SQL Database Restore	Database Backup File	`mytestdb_03122012.bak`
	Download Target Destination	`\\WIN-DOMAIN-CTRL\Backups`
Gather Advanced Parameters for MS SQL Database Restore	Windows Share Password	WinSharePwd **Tip:** To avoid having to re-enter passwords whenever they change, you can create a policy to provide them to the workflow.
	Windows Share User	WIN\Administrator

Be sure that the default values for all remaining parameters are appropriate for your environment (see "Parameters for Restore MS SQL Database" on page 57).

Scenario 2: Restore from a Backup File that is Encrypted and Compressed

This scenario requires you to specify the encryption password for the database backup file. The workflow automatically handles the compression, so there is no need to specify the compression parameter. In this example, the backup file is stored locally on the server where the target instance resides.

Step Name	Parameter Name	Example Value
Gather Parameters for MS SQL Database Restore	Database Backup File	`c:\Backups\mytestdb_03122012.bak`

Step Name	Parameter Name	Example Value
Gather Advanced Parameters for MS SQL Database Restore	Backup Encryption Password	EncryptMyBackup

Be sure that the default values for all remaining parameters are appropriate for your environment (see "Parameters for Restore MS SQL Database" on page 57).

Scenario 3: Overwrite an Existing Database, Restore Users, and Configure Windows Domain User Using Runtime Parameters

This scenario overwrites an existing database and restores any existing users after the restore is performed. In this example, the backup file is stored locally on the server where the target database resides.

> **Note:** You may want to run this workflow against a MS SQL instance that can only be accessed by a Windows user with a temporary password. By using a runtime parameter for the password, you can ensure that the password used is always the latest.
>
> To specify the Windows domain user at the time you execute a deployment with runtime parameters, perform the following additional steps:
>
> 1. When you make a copy of the workflow, expand the appropriate step, and then set the Windows domain user parameters—Instance Account and Instance Password—to **- User selected -**.
> 2. When you create a deployment from the copy of the workflow, set the parameter types to **Runtime Value**.
> 3. When you execute the deployment, specify the Windows domain user account and password.

Step Name	Parameter Name	Example Value
Gather Parameters for MS SQL Database Restore	Database Backup File	`c:\Backups\mytestdb_03122012.bak`
Gather Advanced Parameters for MS SQL Database Restore	Overwrite Existing Database	YES
	Preserve Users and Roles	YES
	Instance Account	Domain\DomainUserAcct **Note:** Enter at runtime.

Step Name	Parameter Name	Example Value
	Instance Password	DomainUserPswd **Note:** Enter at runtime.

Be sure that the default values for all remaining parameters are appropriate for your environment (see "Parameters for Restore MS SQL Database" on the next page).

Parameters for Restore MS SQL Database

The following tables describe the required and optional input parameters for this workflow. Most of these parameters are not initially visible in a deployment. For most parameters, if you do not specify a value for a parameter, a default value is assigned

Parameters Defined in this Step: Gather Parameters for MS SQL Database Restore

Parameter Name	Default Value	Required	Description
Database Backup File	no default	required	Path where the database backup file is (or will be) stored, either locally or on a network share. If the file already exists locally or on a network share, specify the file name in this parameter and the path in the Download Target Destination parameter. If the file does not yet exist locally or on a network share, it will be downloaded into this location from the software repository. If the file is (or will be) stored on a network share, the Windows Share User must have read and write access to that share.
Download Target Destination	no default	required	The directory where the database backup file will be stored. If the database backup file does not yet exist in this directory, it will be downloaded from the software repository and stored in this directory.

Additional Parameters Defined in this Step: Gather Advanced Parameters for MS SQL Database Restore

Parameter Name	Default Value	Required	Description
Backup Encryption Password	no default	optional	To decrypt a backup file that was encrypted with a password, specify the password in this parameter.
Data File Locations	no default	optional	Comma-delimited list of directories or full file paths for each data file in the backup file. Use Run Simulation Only mode to discover the number of data files in the backup file. If this parameter is not specified, the original data file names and paths will be used.
Database Name	no default	optional	To restore the database from the backup file using a different database name, specify that name here. If this parameter is not specified, the original database name will be used.
Instance Account	no default	optional	The Windows account that will perform the restore operation.

Additional Parameters Defined in this Step: Gather Advanced Parameters for MS SQL Database Restore , continued

Parameter Name	Default Value	Required	Description
Instance Password	no default	optional	The password for the Windows account that will perform the restore operation.
Log File Locations	no default	optional	Comma-delimited list of directories or full file paths for each log file in the backup file. Use Run Simulation Only mode to discover the number of log files in backup file. If this parameter is not specified, the original log file names and paths will be used.
Overwrite Existing Database	NO	optional	If set to YES, and the database already exists, the workflow will overwrite the database. Valid values: YES or NO. If set to NO, and the database already exists, the workflow will fail.
Preserve Users and Roles	NO	optional	If set to YES, and the database already exists, the workflow will try to preserve the database users and role. Valid values: YES or NO.
Reindex Restored Database	NO	optional	If set to YES, the workflow will re-index the database after the restore operation is successfully completed. Valid values: YES or NO. Re-indexing improves database performance. More specifically , it recreates all the table look-ups and performance tunes them according to the new environment. This is important when you are restoring a database in a new environment that it has never seen before.
Run Simulation Only	NO	optional	If set to YES, the workflow will only run the Pre-Restore Validation step. It will not attempt to restore the database. Use this mode to discover the original data and log files used for the database backup. Valid values: YES or NO.

MS SQL - Backup and Restore Database

This workflow enables you to backup the contents of a SQL Server database (the **source database**) into a file and restore a database in another instance (the **target instance**) using the contents of that backup file. The source database and target instance are specified at run time.

This is a **bridged execution** workflow. The first group of steps performs the backup on the specified source database. The second group of steps performs the restore on the specified database in the specified target instance.

You can specify various options, including whether the backup file is compressed or encrypted with a password.

> **Note:** Bridged execution workflows work on one target level (server, instance, or database). This workflow runs on the database level at all times. When choosing a target instance at run time, you will actually see a list of databases that reside on each instance. You can select any database in the target instance where you want to perform the restore.
>
> If you specify the RESTORE - Database Name parameter, the workflow will use that database. If you do not specify the RESTORE - Database Name parameter, the workflow will use the original database name from the backup.

If the database specified in the Database Name parameter does not exist in the target instance, the workflow will create it. If the database already exists, you can specify whether you want the workflow to overwrite its contents. You can also specify whether existing database users should be re-created after the restore operation—in which case, any users included in the backup file are ignored .

This workflow also provides a "simulation mode" where the Run MS SQL Pre-Restore Validation step is executed, but the restore is not performed. This is useful for testing or troubleshooting your parameter values.

The workflow performs extensive validation checks prior to and immediately after both the backup and restore operations to ensure that both the backup file and the restored database are valid.

See "Parameters for Backup and Restore MS SQL Database" on page 71 for a list of backup and restore options that you can specify. Many of these parameters are hidden by default

The process of deploying and running this workflow is the same for all scenarios, but the parameters required will differ depending on the specific scenario that you are implementing.

The workflow provides default values for most parameters. These default values are usually sufficient for a "typical" database refresh. You can override the defaults by specifying parameter values in the deployment. You can also expose additional parameters in the workflow, if necessary, to accomplish more advanced scenarios.

Topic	Information Included
"Prerequisites for this Workflow"	List of prerequisites that must be satisfied before you can run this workflow
"How this Workflow Works"	Information about what the workflow does, including validation checks performed, steps executed, and a high-level process flow
"How to Run this Workflow"	Instructions for running this workflow in your environment
"Sample Scenarios"	Examples of typical parameter values for this workflow
"Parameters for Backup and Restore MS SQL Database"	List of input parameters for this workflow

The process of deploying and running this workflow is the same for all scenarios, but the parameters required will differ depending on the specific scenario that you are implementing.

The workflow provides default values for most parameters. These default values are usually sufficient for a "typical" database backup and restore. You can override the defaults by specifying parameter values in the deployment. You can also expose additional parameters in the workflow, if necessary, to accomplish more advanced scenarios.

Any parameters not explicitly specified in the deployment will have the default values listed in "Parameters for Backup and Restore MS SQL Database" on page 71 .

Prerequisites for this Workflow

The following prerequisites must be satisfied before you can run the "MS SQL - Restore Database" workflow:

1. The service login for the SQL Server service must have read and write permissions on the location where the backup file will be stored.
2. The server management agent must have login access to the target SQL Server instance. It must also have permission to create a new database and perform database consistency check (DBCC) commands on the restored database.
3. There must be sufficient space available to create the backup file and restore the database (including both data and logs). The workflow checks for this, and will fail if sufficient space is not available.

Additional Considerations

For information about prerequisites for SQL Server, refer to the SQL Server Product Documentation.

How this Workflow Works

This topic contains the following information about the "MS SQL - Backup and Restore Database" workflow:

Validation Checks Performed

The workflow checks the following things prior to dumping the database. If any of these checks fails, the workflow fails.

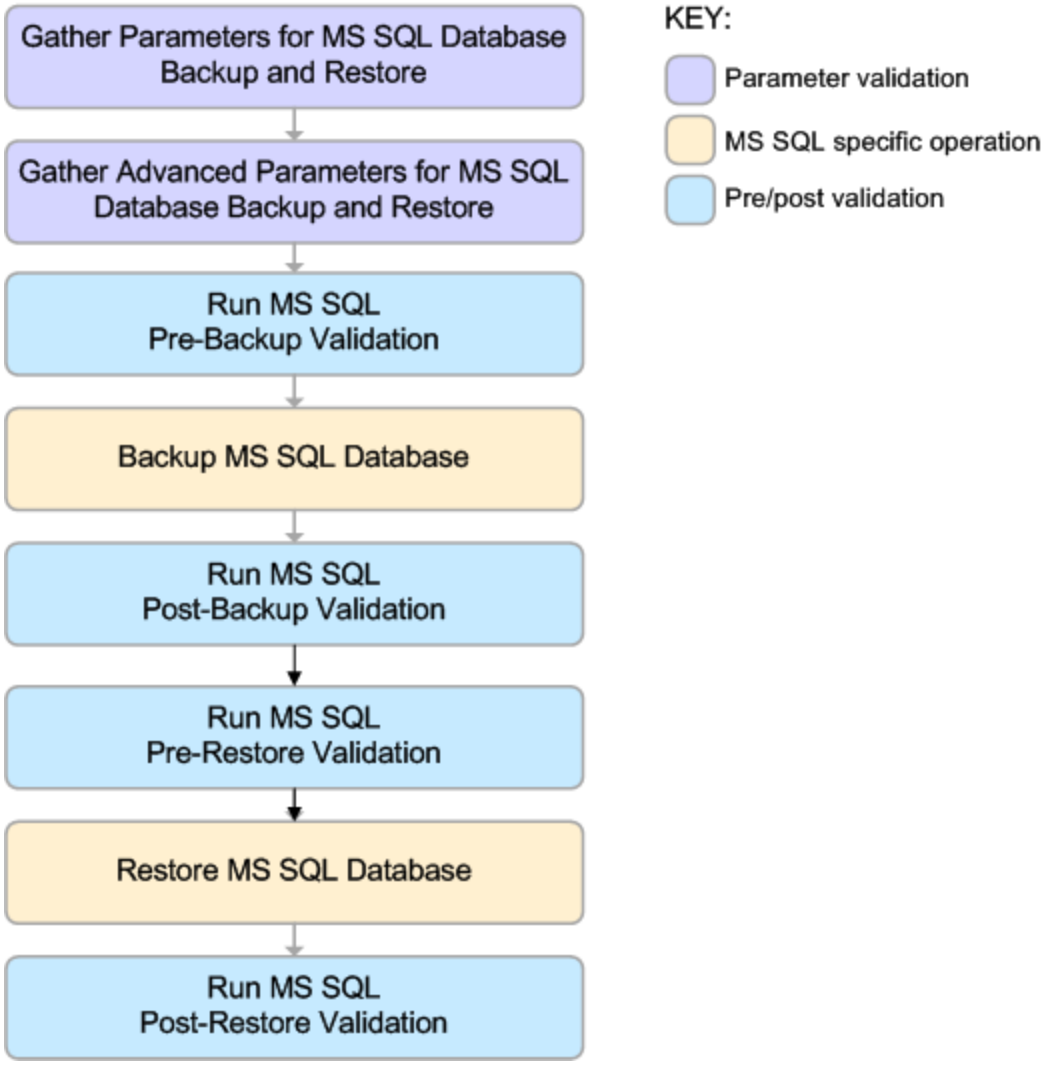
1. All required parameters have values. If any required parameter does not have a value—either a value that you specify or a default value—the workflow fails in either the Run MS SQL Pre-Backup Validation step or the Run MS SQL Pre-Restore Validation step.
2. The Working Path is accessible, either locally or on a network share.

   If the Working Path is on a network share, the BACKUP - Windows Share User has read and write access the share.
3. The source database is compatible with the target instance.
4. If the RESTORE - Database Name parameter is specified, this database name complies with SQL Server database naming conventions.
5. The target instance exists, and the workflow can connect to it.
6. Adequate disk space is available to backup and restore the data and log files.

Steps Executed

The "MS SQL - Backup and Restore Database" workflow includes the following steps. Each step must complete successfully before the next step can start. If a step fails, the workflow reports a failure, and all subsequent steps are skipped.

Click each box in the diagram to view additional information about that step in a new window.



Process Flow

This workflow performs the following tasks:

1. Performs the preliminary validation checks described above.
2. If RESTORE - Preserve Users and Roles was set to YES, creates the Roles Creation and Users Creation scripts.
3. Performs the database backup operation to create the backup file.
4. Performs post-backup validation checks to ensure that all required parameters had valid values.
5. If BACKUP - Perform Integrity Check was set to YES, performs an integrity check on the backup file.
6. If not in simulation mode, performs the database restore operation to load the contents of the backup file.
7. Performs post-restore validation checks to ensure that the restored database is sound.
8. If RESTORE - Preserve Users and Roles was set to YES, re-creates any existing database users and roles.
9. If RESTORE - Reindex Restored Database was set to YES, re-indexes the database.

Tips and Best Practices

It is good practice to run basic database consistency checks (DBCCs) on the source database before you create the database backup to ensure that there are no internal errors in the database.

If you find errors in the source database, be sure to fix them before you run this workflow. This workflow does not have the ability to diagnose or remediate problems in the database prior to performing the database backup.

How to Run this Workflow

This topic explains how to customize and run the "MS SQL - Backup and Restore Database" workflow in your environment.

> **Note:** Prior to running this workflow, review the "Prerequisites for this Workflow", and ensure that all requirements are satisfied.

To customize and run the Backup and Restore MS SQL Database workflow:

1. Create a deployable copy of the workflow.
2. Determine the values that you will specify for the following parameter. This is the minimum set of parameters required to run this workflow.

Parameter Name	Default Value	Description
Working Path	no default	The directory where the database backup file will be stored. This can be a directory or a full file path. This path must be accessible to both the source and target servers. Be sure to specify this parameter in network share notation (for example: `\\<network share>\`). A network path can be located on a target server, but it should always referenced using network share notation instead of local folder notation (for example: `C:\<folder>`). You specify this parameter in the deployment.

   See "Parameters for Backup and Restore MS SQL Database" on page 71 for detailed descriptions of all input parameters for this workflow, including default values.
3. In the workflow editor, expose any additional parameters that you need. You will specify values for these parameters when you create the deployment or at runtime.
4. Save the changes to the workflow (click **Save** in the lower right corner).
5. Create a new deployment.
   a. On the Targets tab, select all the target servers—both source and destination—that will participate in this database refresh. The targets that you select here will be available in the

Target Parameters drop-down menus on the Run page (see step 7).

b. On the Parameters tab, specify values (or set the type to Runtime Value) for the required parameters listed in step 2 and any additional parameters that you exposed in step 3.You do not need to specify values for those parameters whose default values are appropriate for your environment.

6. Save the deployment (click **Save** in the lower right corner).
7. Run the workflow using this deployment, specifying any runtime parameters .

   On the Run page, select the following targets from the respective drop-down menus:

Parameter Name	Default	Description
Source Database	no default	The database from which the backup file will be created. You specify this parameter at run time.
Target Instance	no default	The instance where the database will be restored from the backup file. You specify this parameter at run time. **Note:** Bridged execution workflows work on one target level (server, instance, or database). This workflow runs on the database level at all times. When choosing a target instance at run time, you will actually see a list of databases that reside on each instance. You can select any database in the target instance where you want to perform the restore. If you specify the RESTORE - Database Name parameter, the workflow will use that database. If you do not specify the RESTORE - Database Name parameter, the workflow will use the original database name from the backup.

The workflow will complete and report “Success” on the Console if it has run successfully. If an invalid parameter value is specified, an error is logged, and the workflow terminates in the “Failure” state.

Sample Scenarios

This topic shows you how to use various parameters to achieve the following database backup scenarios in your environment using the "MS SQL - Backup and Restore Database" workflow:

Scenario 1: Backup and Restore Using a Backup File that is Not Encrypted or Compressed

This is the simplest SQL Server database backup and restore scenario. In this example, the backup file is stored on a network share.

Step Name	Parameter Name	Example Value
Gather Parameters for MS SQL Database Backup and Restore	Source Database	Specified at run time.
	Target Instance	Specified at run time.
	Working Path	`\\WIN-DOMAIN-CTRL\Backups`

Scenario 2: Backup and Restore—Overwrite Existing Database and Preserve Existing Users

This scenario requires you to specify the two restore parameters that instruct the workflow to overwrite the existing database and then re-create existing users and roles. In this example, the backup file is stored on a network share.

Note that the BACKUP - Windows Share User and BACKUP - Windows Share Password are specified. This is not required, but it facilitates the disk space check on the network path. If you do not specify this parameter, this check is skipped.

Step Name	Parameter Name	Example Value
Gather Parameters for MS SQL Database Backup and Restore	Source Database	Selected at run time.
	Target Instance	Selected at run time.
	Working Path	`\\WIN-DOMAIN-CTRL\Backups`
Gather Advanced Parameters for MS SQL Database Backup and Restore	BACKUP - Windows Share User	WIN\Administrator
	BACKUP - Windows Share Password	WinSharePwd
	RESTORE - Overwrite Existing Database	YES
	RESTORE - Preserve Users and Roles	YES

Scenario 3: Perform a Backup, Simulate a Restore, and Configure Windows Domain User Using Runtime Parameters

This scenario overwrites an existing database and restores any existing users after the restore is performed. In this example, the backup file is stored on a network share.

> **Note:** You may want to run this workflow against a MS SQL instance that can only be accessed by a Windows user with a temporary password. By using a runtime parameter for the password, you can ensure that the password used is always the latest.
>
> To specify the Windows domain user at the time you execute a deployment with runtime parameters, perform the following additional steps:
>
> 1. When you make a copy of the workflow, expand the appropriate step, and then set the following Windows domain user parameters to **- User selected -**:
>
>   BACKUP - Instance Account
>   BACKUP - Instance Password
>   RESTORE - Instance Account
>   RESTORE - Instance password
>
> 2. When you create a deployment from the copy of the workflow, set the parameter types to **Runtime Value**.
> 3. When you execute the deployment, specify the Windows domain user account and password.

Step Name	Parameter Name	Example Value
Gather Parameters for MS SQL Database Backup and Restore	Source Database	Selected at run time.
	Target Instance	Selected at run time.
	Working Path	`\\WIN-DOMAIN-CTRL\Backups`
Gather Advanced Parameters for MS SQL Database Backup and Restore	BACKUP - Windows Share User	WIN\Administrator
	BACKUP - Windows Share Password	WinSharePwd **Tip:** To avoid having to re-enter passwords whenever they change, you can create a policy to provide them to the workflow.

Step Name	Parameter Name	Example Value
	ALL - Run Simulation Only	YES
	BACKUP - Instance Account	Domain\DomainUserAcct **Note:** Enter at runtime.
	BACKUP - Instance Password	DomainUserPswd **Note:** Enter at runtime.
	RESTORE - Instance Account	Domain\DomainUserAcct **Note:** Enter at runtime.
	RESTORE - Instance Password	DomainUserPswd **Note:** Enter at runtime.

Be sure that the default values for all remaining parameters are appropriate for your environment (see Parameters for Backup and Restore MS SQL Database).

Parameters for Backup and Restore MS SQL Database

The following tables describe the required and optional input parameters for this workflow. Most of these parameters are not initially visible in a deployment. For most parameters, if you do not specify a value for a parameter, a default value is assigned

Parameters Defined in this Step: Gather Parameters for Backup and Restore MS SQL Database

Parameter Name	Default Value	Required	Description
Source Database	no default	required	The database from which the backup file will be created. You specify this parameter at run time.
Target Instance	no default	required	The instance where the database will be restored from the backup file. You specify this parameter at run time. **Note:** Bridged execution workflows work on one target level (server, instance, or database). This workflow runs on the database level at all times. When choosing a target instance at run time, you will actually see a list of databases that reside on each instance. You can select any database in the target instance where you want to perform the restore. If you specify the RESTORE - Database Name parameter, the workflow will use that database. If you do not specify the RESTORE - Database Name parameter, the workflow will use the original database name from the backup.
Working Path	no default	required	The directory where the database backup file will be stored. This can be a directory or a full file path. This path must be accessible to both the source and target servers. Be sure to specify this parameter in network share notation (for example: `\\<network share>\`). A network path can be located on a target server, but it should always referenced using network share notation instead of local folder notation (for example: `C:\<folder>`). You specify this parameter in the deployment.

Additional Parameters Defined in this Step: Gather Advanced Parameters for Backup and Restore MS SQL Database

Parameter Name	Default Value	Required	Description
ALL - Encryption Password	no	optional	Password used to encrypt

Additional Parameters Defined in this Step: Gather Advanced Parameters for Backup and Restore MS SQL Database, continued

Parameter Name	Default Value	Required	Description
	default		and decrypt the backup file. To decrypt a backup file that was encrypted with a password, specify the password in this parameter.
ALL - Run Simulation Only	No	optional	If set to YES, the workflow will only run the Pre-Restore Validation step. It will not attempt to restore the database. Use this mode to discover the original data and log files used for the database backup. Valid values: YES or NO.
BACKUP - Backup Description	no default	optional	Text that describes this backup (up to 255 characters).
BACKUP - Backup Name	no default	optional	The name of this backup (up to 128 characters).
BACKUP - Compress Backup File	NO	optional	If you specify YES, the backup file will be compressed. Valid values: YES or NO. Compression is supported on SQL Server 2008 Enterprise and later.
BACKUP - Expiration Date	no default	optional	Date and time when the backup file expires and the backup data is no longer considered relevant.After this date and time, SQL Server is not prevented from overwriting this backup file. The Expiration Date must

Additional Parameters Defined in this Step: Gather Advanced Parameters for Backup and Restore MS SQL Database, continued

Parameter Name	Default Value	Required	Description
			be specified in a format compatible with the configured system datetime format. If both the Retention Days and the Expiration Date parameters are specified, the Retention Days parameter takes precedence.
BACKUP - Instance Account	no default	optional	The Windows account that will perform the backup operation.
BACKUP - Instance Password	no default	optional	The password for the Windows account that will perform the backup operation.
BACKUP - Perform Integrity Check	NO	optional	If you specify YES, the workflow will perform an integrity check on the database backup file. Valid values: YES or NO.
BACKUP - Retention Days	no default	optional	Number of days that must elapse before this backup file can be overwritten by SQL Server. If both the Retention Days and the Expiration Date parameters are specified, the Retention Days parameter takes precedence.
BACKUP - Windows Share Password	no default	optional	Password for the user specified in Windows Share User.
BACKUP - Windows Share User	no default	optional	Windows user who can access the specified Windows network share

Additional Parameters Defined in this Step: Gather Advanced Parameters for Backup and Restore MS SQL Database, continued

Parameter Name	Default Value	Required	Description
			and who will own (and write) the backup file.
RESTORE - Data File Locations	no default	optional	Comma-delimited list of directories or full file paths for each data file in the backup file. Use Run Simulation Only to discover the number of data files in backup file. If this parameter is not specified, the original data file name will be used.
RESTORE - Database Name	no default	optional	To restore the database from the backup file using a different database name, specify that name here. If this parameter is not specified, the original database name will be used.
RESTORE - Download Target Destination	no default	optional	The directory where the database backup file will be stored.
RESTORE - Instance Account	no default	optional	The Windows account that will perform the restore operation.
RESTORE - Instance Password	no default	optional	The password for the Windows account that will perform the restore operation.
RESTORE - Log File Locations	no default	optional	Comma-delimited list of directories or full file paths for each log file in the backup file. Use Run Simulation Only mode to discover the number of log files in backup file. If this parameter is not specified, the original log file name

Additional Parameters Defined in this Step: Gather Advanced Parameters for Backup and Restore MS SQL Database, continued

Parameter Name	Default Value	Required	Description
			will be used.
RESTORE - Overwrite Existing Database	NO	optional	If set to YES, and the database already exists, the workflow will overwrite the database. Valid values: YES or NO.
RESTORE - Preserve Users and Roles	NO	optional	If set to YES, and the database already exists, the workflow will overwrite the database. Valid values: YES or NO.
RESTORE - Reindex Restored Database	NO	optional	If set to YES, the workflow will re-index the database after the restore operation is successfully completed. Valid values: YES or NO. Re-indexing improves database performance. More specifically , it recreates all the table look-ups and performance tunes them according to the new environment. This is important when you are restoring a database in a new environment that it has never seen before.

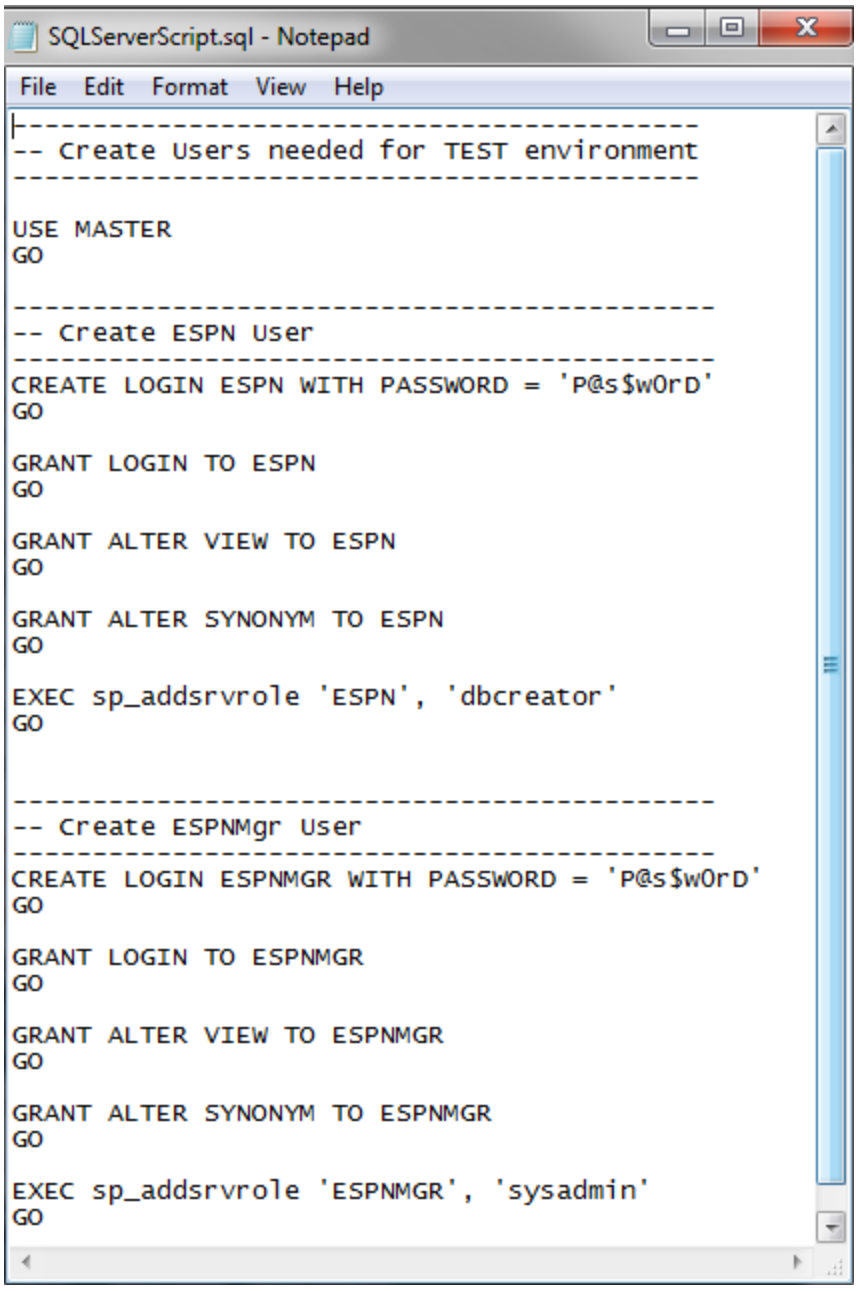
DB Release for SQL Server v2

This workflow will check a list of T-SQL script and embedded SQL files for disallowed commands, check the syntax, then execute the files on the targetMicrosoft SQL Server Microsoft SQL Server databases if they pass all required tests.

This workflow is designed for SQL script transactions to be deployed and executed against target SQL Server databases. SQL scripts are stored and downloaded from the HPE DMA software repository.

If the SQL scripts are embedded within a SQL script, this workflow has the ability to download the embedded scripts from SA core, provided the location of the sub-script is same as the staging directory. This workflow can download only one level of embedded SQL scripts.

Before running the DB Release for SQL Server workflow you need to create the SQL script file (or files). For example:


SQLServerScript.sql - Notepad
File Edit Format View Help
-- Create Users needed for TEST environment
USE MASTER
GO
-- Create ESPN User
CREATE LOGIN ESPN WITH PASSWORD = 'P@s$w0rD'
GO
GRANT LOGIN TO ESPN
GO
GRANT ALTER VIEW TO ESPN
GO
GRANT ALTER SYNONYM TO ESPN
GO
EXEC sp_addsrvrole 'ESPN', 'dbcreator'
GO
-- Create ESPNMgr User
CREATE LOGIN ESPNMGR WITH PASSWORD = 'P@s$w0rD'
GO
GRANT LOGIN TO ESPNMGR
GO
GRANT ALTER VIEW TO ESPNMGR
GO
GRANT ALTER SYNONYM TO ESPNMGR
GO
EXEC sp_addsrvrole 'ESPNMGR', 'sysadmin'
GO

You can customize what the workflow checks in the SQL scripts:

- SQL advanced features
- SQL database commands
- SQL database links
- SQL syntax
- SQL system grants
- A regular expression

If all the tests pass, the SQL scripts may be deployed and executed against the target SQL Server databases.

> **Note:** This workflow does not provide any rollback capability.

To use this workflow in your environment, see the following information:

Topic	Information Included
"Prerequisites for this Workflow"	List of prerequisites that must be satisfied before you can run this workflow
"How this Workflow Works"	Information about what the workflow does, including validation checks performed, and steps executed
"How to Run this Workflow"	Instructions for running this workflow in your environment
"Sample Scenarios"	Examples of typical parameter values for this workflow
Parameters for Oracle - SQL Release v2	List of input parameters for this workflow

Prerequisites for this Workflow

Be sure that the following prerequisites are satisfied before you run the "DB Release for SQL Server v2" workflow.

Dependencies

- This solution requires HPE DMA version 10.40 (or later).

  The latest HPE DMA solution packs require the latest HPE DMA platform. To use the latest solution packs, update the HPE DMA platform. HPE DMA10.50 solution packs are supported on HPE DMA10.50 (and later).
- You have installed the Database Compliance solution pack.
- An SQL Server instance and its databases should already be provisioned and added to the Environment section—this can be accomplished by using Discovery.
- The SQL scripts must be available in the HPE DMA software repository.
- You have installed the `osql` or `SQLCMD` utility and made it accessible via the user/password settings stored in the metadata. Check the Environment page for those settings. If there is no metadata, the connection will use Windows authentication.
- You need an SA (System Administrator) role to perform any server level or database level updates.

Supported Versions of SQL Server

2008, 2008 R2, 2012, 2014

SQL Scripts

You need to create the SQL scripts that manage the release. The files may contain the normal SQL Server DML and DDL commands.

> **Tip:** List the SQL scripts in the SQL scripts parameter in the order in which they need to be executed.

SQL Server Documentation

For more information about prerequisites for SQL Server, refer to the Microsoft SQL Server Documentation.

How this Workflow Works

The following information describes how the "DB Release for SQL Server v2" workflow works.

Overview show

The workflow starts by gathering input parameters.

If the SQL scripts do not exist on the specified target location, they are downloaded from the software repository.

Based on the parameters you set when you create your deployment, the workflow will do the following:

- Check the SQL code for SQL advanced features—unless specified in the exception list. If any are found, the workflow will exit with a failure code.
- Check the SQL code for SQL database commands—unless specified in SQL commands to be excluded from the check. If any are found, the workflow will exit with a failure code.
- Check the SQL code for any SQL database links—if any are found, the workflow will exit with a failure code.
- Check the SQL code for syntax errors—if any are found, the workflow will exit with a failure code.
- Check the SQL code for any SQL system grants—unless specified in the exception list. If any are found, the workflow will exit with a failure code.
- Check the SQL code for a regular expression that you specify—if any matches are found, the workflow will exit with a failure code.

If there were no errors in the checks and the Run Flag is set, the workflow uses the osql or SQLCMD utility to execute the SQL script files.

Validation Checks Performed show

This workflow validates the SQL scripts in the following ways:

1. If you set the Run Flag to Check SQL Advanced Features, the workflow searches for any instance configuration options—unless included in your exclusion list. These are instance level settings that most users shouldn't be changing, for example, startup procs and xp_cmdshell.
2. If you set the Run Flag to Check SQL Database Commands, the workflow searches the SQL statements for the commands that you specify in SQL Commands.
3. If you set the Run Flag to Check SQL Database Links, the workflow searches the SQL statements for OPENQUERY, OPENROWSET, and OPENDATASOURCE statements. It also checks for this pattern: [server].[instance].[owner].[database]
4. If you set the Run Flag to Check SQL Syntax, the workflow verifies that all the SQL statements have valid syntax.
5. If you set the Run Flag to Check SQL System Grants, the workflow searches the SQL statements for any system level (server role) grants—unless included in your exclusion list. For example: GRANT CONTROL SERVER TO SOMEUSER
6. If you set the Run Flag to Match Regular Expression to SQL Server Scripts and you specify a regular expression, the workflow searches the SQL statements for any regex matches.

If any of the validations fail, the workflow will output the offending SQL line to `stdout`, return an error status, and the SQL scripts will not be executed.

Steps Executed show

The "DB Release for SQL Server v2" workflow includes the following steps. Each step must complete successfully before the next step can start. If a step fails, the workflow reports a failure and all subsequent steps are skipped.

Steps Used in DB Release for SQL Server

Workflow Step	Description
MS SQL - Parameters - DB Release for SQL Server	This step accepts the basic input parameters for the workflow. The parameters will be used in subsequent steps.
Check if Download File Exists	This step determines whether one or more specified files already exist on the target server.
Check For Nested SQL files in MSSQL SQL file	This step checks for embedded SQL scripts.
Download Software	This step downloads a list of files to a specified location on the target

Steps Used in DB Release for SQL Server, continued

Workflow Step	Description
	server.
Check SQL Advanced Features	This step checks the SQL scripts for any advanced feature non-default setting. An exception list can be specified to exclude specific advanced features from the check.
Check SQL Database Commands	This step checks the SQL scripts to ensure that specific types of SQL database commands—as specified in the SQL Commands parameter—are not included.
Check SQL Database Links	This step checks an SQL Script for any database link usage.
Check SQL Syntax	This step verifies the syntax of an SQL Server Script. The step assumes that a `go` statement on its own line signifies the end of a code block.
Check SQL System Grants	This step checks an SQL Script for any system level (server role) grants. An exception list can be specified to exclude specific privileges from the check.
Match Regular Expression to SQL Server Scripts	This step applies a regular expression to each SQL statement in an SQL Script file. If any `regex` matches are found, they are output to `stdout` and an error is returned.
Run SQL Server Script v2	This step executes SQL Scripts using osql.exe. This step is only executed if all the previous checks passed.

Note: For input parameter descriptions and defaults, see "Parameters for DB Release for SQL Server v2" on page 92.

How to Run this Workflow

The following instructions show you how to customize and run the "DB Release for SQL Server v2" workflow in your environment.

The workflow provides default values for some parameters. These default values are usually sufficient for a "typical" installation. You can override the defaults by specifying parameter values in the deployment. You can also expose additional parameters in the workflow, if necessary, to accomplish more advanced scenarios. Any parameters not explicitly specified in the deployment will have the default values listed in "Parameters for DB Release for SQL Server v2" on page 92.

> **Note:** Before following this procedure, review the "Prerequisites for this Workflow" on page 79, and ensure that all requirements are satisfied.

To use the DB Release for SQL Server workflow:

1. Create a deployable copy of the workflow.
2. Determine the values that you will specify for the following parameters:

Input Parameters for MS SQL - Parameters - DB Release for SQL Server

Parameter Name	Default Value	Required	Description
Display SQL Length	2000	optional	The number of characters of a SQL batch that is displayed when an error occurs. Enter "0" to display the entire code. **Note:** Displaying the entire code may cause performance issues for your browser.
File List	no default	required	Comma-separated list of the files that contain the SQL scripts that will be checked. **Note:** List the SQL script files in the order in which they need to be executed.
Staging Directory	`C:\Temp\`	optional	The directory that contains the SQL scripts that will be checked.

Input Parameters for Check SQL Advanced Features

Parameter Name	Default Value	Required	Description
Exception List	see	optional	Comma-separated list of

Input Parameters for Check SQL Advanced Features, continued

Parameter Name	Default Value	Required	Description
	description		advanced features that will be allowed. For example, if you specify CURSOR THRESHOLD, QUERY WAIT, those advanced features will be allowed—any other advanced features that occur in the code will cause the workflow to fail. The default is to check all of the normal advanced features.
Run Flag	Y	optional	Flag to indicate whether the workflow should run this check. Valid values are Y (run the check) or N (do not run the check).

Input Parameters for Check SQL Database Commands

Parameter Name	Default Value	Required	Description
Run Flag	Y	optional	Flag to indicate whether the workflow should run this check. Valid values are Y (run the check) or N (do not run the check).
SQL Commands	shutdown, sp_configure, create database, drop database, create login, create user, drop login, drop user, sp_ grantdbaccess, sp_ addlogin, sp_droplogin	optional	Comma-separated list of SQL commands that are not allowed. The default shows an example of how to fill out the list. You may want to customize this list for your configuration.

Input Parameters for Check SQL Database Links

Parameter Name	Default Value	Required	Description
Run Flag	Y	optional	Flag to indicate whether the workflow should run this check. Valid values are Y (run the check) or N (do not run the check).

Input Parameters for Check SQL Syntax

Parameter Name	Default Value	Required	Description
Run Flag	Y	optional	Flag to indicate whether the workflow should run this check. Valid values are Y (run the check) or N (do not run the check).

Input Parameters for Check SQL System Grants

Parameter Name	Default Value	Required	Description
Exception List	grant db_owner, grant ddladmin, grant sysadmin, grant securityadmin,grant serveradmin, grant processadmin, grant diskadmin, grant dbcreator	optional	Comma-separated list of SQL system privileges that will be allowed. For example, if you specify , those system privileges will be allowed—any other system privileges that occur in the code will cause the workflow to fail. The default shows an example of how to fill out the list. You may want to customize this list for your configuration.
Run Flag	Y	optional	Flag to indicate whether the workflow should run this check. Valid values are Y (run the check) or N (do not run the check).

Input Parameters for Match Regular Expression to SQL Server Scripts

Parameter Name	Default Value	Required	Description
Regular Expression		optional	The regular expression to be searched for in all of the SQL scripts to be deployed. If the specified regular expression is found, the workflow exits with a failure. For example: drop\s+table will match all statements that drop a table.
Run Flag	Y	optional	Flag to indicate whether the workflow should run this check. Valid values are Y (run the check) or N (do not run the check).

Input Parameters for Run SQL Server Script

Parameter Name	Default Value	Required	Description
Database Name	master	optional	The name of the database to which the specified SQL scripts will be applied.
Run Flag	Y	optional	Flag to indicate whether the workflow should run the SQL Server script. Valid values are Y (run the check) or N (do not run the check).

> **Note:** See "Parameters for DB Release for SQL Server v2" on page 92 for detailed descriptions of all input parameters for this workflow, including default values.

3. In the workflow editor, expose any additional parameters that you need. You will specify values for those parameters when you create the deployment or at runtime.
4. Save the changes to the workflow (click **Save** in the lower right corner).
5. Create a new deployment.
6. On the Parameters tab, specify values (or set the type to Runtime Value) for the required parameters listed in step 2 and any additional parameters that you have exposed. You do not need to specify values for those parameters whose default values are appropriate for your environment.
7. On the Targets tab, specify one or more targets for this deployment.
8. Save the deployment (click **Save** in the lower right corner).
9. Run the workflow using this deployment, specifying any runtime parameters.

To verify the results:

The workflow will complete and report SUCCESS on the Console if it has run successfully. If an error occurs during workflow execution, the error is logged, and the workflow terminates in the FAILURE state.

Optional: If you want to further verify the results:

Log in to your database to make sure that whatever you created or modified was actually done.

To view the output:

The workflow writes the execution output for SQL script execution in the HPE DMA Steplog.

Sample Scenarios

This topic shows you typical parameter values for different use cases for the "DB Release for SQL Server v2" workflow.

Scenario 1: Check the SQL script files for disallowed commands, check the syntax, then deploy and execute the scripts show

You only need to specify the File List and the Staging Directory since this scenario takes advantage of many parameter defaults. The workflow will check the SQL script files for:

- All of the normal advanced features
- All of the SQL database commands that are in the default SQL Commands parameter
- SQL database links
- SQL syntax
- All the SQL system grants—except those in the default Exception List parameter
- No regular expression—since none is specified by default

As long as no error is discovered in the checks, the SQL scripts will be deployed and executed on the target SQL Server databases.

Determine the values that you will specify for the following parameters:

Input Parameters for MS SQL - Parameters - DB Release for SQL Server

Parameter Name	Example Value	Description
File List	sqlserverscript.sql	Comma-separated list of the files that contain the SQL scripts that will be checked. **Note:** List the SQL script files in the order in which they need to be executed.
Staging Directory	`C:\Temp\`	The directory that contains the SQL scripts that will be checked.

Be sure that the default values for all remaining input parameters are appropriate for your environment (see "Parameters for DB Release for SQL Server v2" on page 92).

Scenario 2: Check the SQL script files for disallowed commands, check the syntax, configure Windows domain user using runtime parameters, but do not deploy and execute the scripts show

This scenario takes advantage of many parameter defaults and also demonstrates some optional parameters. The workflow will check the SQL script files for:

- All of the SQL database commands that are in the default SQL Commands parameter
- SQL database links
- SQL syntax
- All the SQL system grants—except those in the default Exception List parameter
- The regular expression drop\s+table
- A database to which the SQL scripts will be applied

> **Note:** You may want to run this workflow against a MS SQL instance that can only be accessed by a Windows user with a temporary password. By using a runtime parameter for the password, you can ensure that the password used is always the latest.
>
> To specify the Windows domain user at the time you execute a deployment with runtime parameters, perform the following additional steps:
>
> 1. When you make a copy of the workflow, expand the appropriate step, and then set the Windows domain user parameters—Instance Account and Instance Password—to **- User selected -**.
> 2. When you create a deployment from the copy of the workflow, set the parameter types to **Runtime Value**.
> 3. When you execute the deployment, specify the Windows domain user account and password.

This workflow run will only report the results of the checks. The SQL scripts will NOT be deployed and executed on the target SQL Server databases.

Determine the values that you will specify for the following parameters:

Input Parameters for MS SQL - Parameters - DB Release for SQL Server

Parameter Name	Example Value	Description
File List	sqlserverscript.sql	Comma-separated list of the files that contain the SQL scripts that will be checked.

Input Parameters for MS SQL - Parameters - DB Release for SQL Server, continued

Parameter Name	Example Value	Description
		Note: List the SQL script files in the order in which they need to be executed.
Instance Account	Domain\DomainUserAcct **Note:** Enter at runtime.	The Windows account that will perform the release management.
Instance Password	DomainUserPswd **Note:** Enter at runtime.	The password for the Windows account that will perform the release management.
Staging Directory	`C:\Temp\`	The directory that contains the SQL scripts that will be checked.

Input Parameters for Match Regular Expression to SQL Server Scripts

Parameter Name	Example Value	Description
Regular Expression	drop\s+table	The regular expression to be searched for in all of the SQL scripts to be deployed. If the specified regular expression is found, the workflow exits with a failure. For example: drop\s+table will match all statements that drop a table.
Run Flag	Y	Flag to indicate whether the workflow should run this check. Valid values are Y (run the check) or N (do not run the check).

Input Parameters for Run SQL Server Script

Parameter Name	Example Value	Description
Database Name	mydb	The name of the database to which the specified SQL scripts will be applied.
Run Flag	N	Flag to indicate whether the workflow should run the SQL Server script. Valid values are Y (run the check) or N (do not run the check).

Note: Some of these parameters are not exposed by default in the deployment.

Be sure that the default values for all remaining input parameters are appropriate for your environment (see "Parameters for DB Release for SQL Server v2" on the next page).

Parameters for DB Release for SQL Server v2

The following tables describe the required and optional input parameters for this workflow. Some of these parameters may not be initially visible in a deployment. For some parameters, if you do not specify a value for a parameter, a default value is assigned.

> **Note:** Only those parameters that are configurable in a standard deployment are listed here. Input parameters that must be mapped to output parameters of previous steps are not listed.

Input Parameters Defined in this Step: MS SQL - Parameters - DB Release for SQL Server

Parameter Name	Default Value	Required	Description
Display SQL Length	2000	optional	The number of characters of a SQL batch that is displayed when an error occurs. Enter "0" to display the entire code. **Note:** Displaying the entire code may cause performance issues for your browser.
File List	no default	required	Comma-separated list of the files that contain the SQL scripts that will be checked. **Note:** List the SQL script files in the order in which they need to be executed.
Instance Account	no default	optional	The Windows account that will perform the release management.
Instance Password	no default	optional	The password for the Windows account that will perform the release management.
Staging Directory	`C:\Temp\`	optional	The directory that contains the SQL scripts that will be checked.

Additional Input Parameters Defined in this Step: Check SQL Advanced Features

Parameter Name	Default Value	Required	Description
Exception List	see description	optional	Comma-separated list of advanced features that will be allowed. For example, if you specify CURSOR THRESHOLD, QUERY WAIT, those advanced features will be allowed—any other advanced features that occur in the code will cause the workflow to fail. The default is to check all of the

Additional Input Parameters Defined in this Step: Check SQL Advanced Features, continued

Parameter Name	Default Value	Required	Description
			normal advanced features.
Run Flag	Y	optional	Flag to indicate whether the workflow should run this check. Valid values are Y (run the check) or N (do not run the check).

Additional Input Parameters Defined in this Step: Check SQL Database Commands

Parameter Name	Default Value	Required	Description
Run Flag	Y	optional	Flag to indicate whether the workflow should run this check. Valid values are Y (run the check) or N (do not run the check).
SQL Commands	shutdown, sp_configure, create database, drop database, create login, create user, drop login, drop user, sp_ grantdbaccess, sp_ addlogin, sp_droplogin	optional	Comma-separated list of SQL commands that are not allowed. The default shows an example of how to fill out the list. You may want to customize this list for your configuration.

Additional Input Parameters Defined in this Step: Check SQL Database Links

Parameter Name	Default Value	Required	Description
Run Flag	Y	optional	Flag to indicate whether the workflow should run this check. Valid values are Y (run the check) or N (do not run the check).

Additional Input Parameters Defined in this Step: Check SQL Syntax

Parameter Name	Default Value	Required	Description
Run Flag	Y	optional	Flag to indicate whether the workflow should run this check. Valid values are Y (run the check) or N (do not run the check).

Additional Input Parameters Defined in this Step: Check SQL System Grants

Parameter Name	Default Value	Required	Description
Exception List	grant db_owner, grant ddladmin, grant sysadmin, grant securityadmin,grant serveradmin, grant processadmin, grant diskadmin, grant dbcreator	optional	Comma-separated list of SQL system privileges that will be allowed. For example, if you specify , those system privileges will be allowed—any other system privileges that occur in the code will cause the workflow to fail. The default shows an example of how to fill out the list. You may want to customize this list for your configuration.
Run Flag	Y	optional	Flag to indicate whether the workflow should run this check. Valid values are Y (run the check) or N (do not run the check).

Additional Input Parameters Defined in this Step: Match Regular Expression to SQL Server Scripts

Parameter Name	Default Value	Required	Description
Regular Expression		optional	The regular expression to be searched for in all of the SQL scripts to be deployed. If the specified regular expression is found, the workflow exits with a failure. For example: drop\s+table will match all statements that drop a table.
Run Flag	Y	optional	Flag to indicate whether the workflow should run this check. Valid values are Y (run the check) or N (do not run the check).

Additional Input Parameters Defined in this Step: Run SQL Server Script

Parameter Name	Default Value	Required	Description
Database Name	master	optional	The name of the database to which the specified SQL scripts will be applied.
Run Flag	Y	optional	Flag to indicate whether the workflow should run the SQL Server script. Valid values are Y (run the check) or N (do not run the check).

MS SQL Drop Database

This workflow is supported on the Windows operating system platform. The MS SQL Drop Database enables you to remove the target database from the MS SQL instance and from the DMA environment..

To use this workflow in your environment, see the following information:

Topic	Information Included
"Prerequisites for this Workflow" on the next page	List of prerequisites that must be satisfied before you can run this workflow
"How this Workflow Works" on page 97	Information about what the workflow does, including validation checks performed, steps executed, and step descriptions
"How to Run this Workflow" on page 98	Instructions for running this workflow in your environment
"Parameters for MS SQL - Drop Database" on page 100	List of input parameters for this workflow

Prerequisites for this Workflow

Be sure that the following prerequisites are satisfied before you run the MS SQL Drop Database workflow:

- This solution requires HPE DMA version 10.30 (or later).

  The latest HPE DMA solution packs require the latest HPE DMA platform. To use the latest solution packs, update the HPE DMA platform. HPE DMA10.50 solution packs are supported on HPE DMA10.50 (and later).
- You have installed the Database Compliance solution pack.

The workflow must be able to:

- MS SQL service should be up and running.
- Log in to the MS SQL instance using MS SQL login credentials.
- It should drop the database upon connecting to the MS SQL instance.

The information presented here assumes the following: show assumptions

- HPE DMA is installed and operational.
- At least one suitable target server is available.
- You are logged in to the HPE DMA web interface.
- You have permission to create, edit, and deploy copies of the workflows included in this solution pack.

For more information about prerequisites for MS SQL database, refer to the MS SQL Server Documentation.

How this Workflow Works

This workflow performs the following actions:

Drops a MS SQL database and removes it from the DMA environment.

Steps Executed by the Workflow

The MS SQL Drop Database workflow includes the following steps. Each step must complete successfully before the next step can start. If a step fails, the workflow reports a failure and all subsequent steps are skipped.

Steps Used by MS SQL

Workflow Step	Description
Gather Advanced Parameters for MS SQL Drop Database	This step gathers the parameters required to drop an MS SQL database.
MS SQL Check Database Exists	This step validates the existence of the database. Access to the master database is required for validation.
MS SQL Kill Processes	This step kills all the currently running user processes on the target database.
MS SQL Drop Database	This step drops the target database. To run this step, ensure that there are no active connections prior to running this step by running the "MS SQL: Kill Processes" step.
MS SQL Check Database Exists	This step validates the existence of a database. Access to the master database is required for validation.
Remove Database from Environment V2	This step removes the database from the DMA environment. This step takes the Instance Name and Database Name as input parameters. If the Instance Name and Database Name are not provided as input parameters, then the database against which the workflow is being executed will be removed from the DMA environment.

> **Note:** For input parameter descriptions and defaults, see "Parameters for MS SQL - Drop Database" on page 100.

How to Run this Workflow

The following instructions show you how to customize and run the MS SQL Drop Database workflow in your environment.

The workflow provides default values for some parameters. These default values are usually sufficient for a "typical" installation. You can override the defaults by specifying parameter values in the deployment. You can also expose additional parameters in the workflow, if necessary, to accomplish more advanced scenarios. Any parameters not explicitly specified in the deployment will have the default values listed in "Parameters for MS SQL - Drop Database" on page 100.

> **Note:** Before following this procedure, review the "Prerequisites for this Workflow" on page 96, and ensure that all requirements are satisfied.

To use the Run MySQL Drop Database workflow:

1. Create a deployable copy of the workflow.
2. Determine the values that you will specify for the following parameters:

   > **Note:** There are no mandatory parameters required to run this workflow. All parameters are optional. You may need to expose additional parameters depending on your objectives.
   >
   > See "Parameters for MS SQL - Drop Database" on page 100 for detailed descriptions of all input parameters for this workflow, including default values.

3. In the workflow editor, expose any additional parameters that you need. You will specify values for those parameters when you create the deployment or at runtime.
4. Save the changes to the workflow (click **Save** in the lower right corner).
5. Create a new deployment.
6. On the Parameters tab, specify values (or set the type to Runtime Value) for the required parameters listed in step 2 and any additional parameters that you have exposed. You do not need to specify values for those parameters whose default values are appropriate for your environment.
7. On the Targets tab, specify one or more targets for this deployment.
8. Save the deployment (click **Save** in the lower right corner).
9. Run the workflow using this deployment, specifying any runtime parameters.

To verify the results:

The workflow will complete and report SUCCESS on the Console if it has run successfully. If an error occurs during workflow execution, the error is logged, and the workflow terminates in the FAILURE state. Also verify by checking that the target database no longer appears in the DMA Environment section.

Parameters for MS SQL - Drop Database

There are no mandatory parameters required to run this workflow. All parameters are optional. Some of these parameters are not initially visible in a deployment. For most parameters, if you do not specify a value for a parameter, a default value is assigned.

MS SQL - Upgrade Standalone SQL Instance

This workflow is supported on the Windows operating system platform. The MS SQL - Upgrade Standalone SQL Instance enables you to update and existing instance of SQL Server.

To use this workflow in your environment, see the following information:

Topic	Information Included
"Prerequisites for this Workflow" on the next page	List of prerequisites that must be satisfied before you can run this workflow
"How this Workflow Works" on page 102	Information about what the workflow does, including validation checks performed, steps executed, and step descriptions
"How to Run this Workflow" on page 104	Instructions for running this workflow in your environment
"Parameters for MS SQL - Upgrade Standalone SQL Instance" on page 106	List of input parameters for this workflow

Prerequisites for this Workflow

Be sure that the following prerequisites are satisfied before you run the MS SQL - Upgrade Standalone SQL Instance workflow:

- This solution requires HPE DMA version 10.30 (or later).

  The latest HPE DMA solution packs require the latest HPE DMA platform. To use the latest solution packs, update the HPE DMA platform. HPE DMA10.50 solution packs are supported on HPE DMA10.50 (and later).
- You have installed the Database Compliance solution pack.

The workflow must be able to:

- MS SQL service should be up and running.
- Log in to the MS SQL instance using MS SQL login credentials.
- It should drop the database upon connecting to the MS SQL instance.

The information presented here assumes the following: show assumptions

- HPE DMA is installed and operational.
- At least one suitable target server is available.
- You are logged in to the HPE DMA web interface.
- You have permission to create, edit, and deploy copies of the workflows included in this solution pack.

For more information about prerequisites for MS SQL database, refer to the MS SQL Server Documentation.

How this Workflow Works

This workflow performs the following actions:

Upgrades an existing standalone instance of SQL Server 2005/08/08R2/12 to SQL Server 2008/08R2/12/14 on an existing Windows 2008/08R2/12/12 R2 server.

Steps Executed by the Workflow

The MS SQL - Upgrade Standalone SQL Instance workflow includes the following steps. Each step must complete successfully before the next step can start. If a step fails, the workflow reports a failure and all subsequent steps are skipped.

Steps Used by MS SQL

Workflow Step	Description
MS SQL - Parameters - Upgrade Standalone	This step gathers all the required parameters for a standalone SQL Server upgrade.
MS SQL - Advanced Parameters - Upgrade Standalone	This step gathers all the optional parameters for a standalone SQL Server upgrade. All advanced parameters are hidden in the deployment screen by default. In order to activate an advanced parameter, go into the Workflow, and change the parameter mapping from on this step from Blank to User Input.
Check If Download File Exists	This step is designed to facilitate the complicated methodologies that various companies use to distribute their software bundles for installation.
MS SQL - Create Install or Upgrade Template	This step verifies that all required parameters are provided, and writes any optional parameters to the template file if they are non-blank.
Unzip Archive	This step is to unzip a zip archive, verify if the input file exists, ensure the output directory exists, creates required directories, and deploys archived files.
MS SQL - Simulate - Install or Upgrade	This step verifies that all required parameters are provided, and the system meets minimum requirements.
MS SQL - Install or Upgrade	This step installs SQL Server 2008 by running the setup.exe program located on the installation media.
MS SQL	This step determines if the target instance name of SQL Server is currently installed.

Steps Used by MS SQL , continued

Workflow Step	Description
Verify SQL Installation	
Windows Check for Pending Reboot	This step checks for any pending reboots.
Discover SQL Databases	This step audits the server's physical environment looking for SQLServer instances and databases.
Windows Restart Server	This step restarts a system.
Windows Wait for Restart	This step is to wait 8 minutes for Windows server to finish restart.
MS SQL - Install or Upgrade	This installs SL Server 2008 by running the setup.exe program located on the installation media.
MS SQL Verify SQL Installation	This step determines if the target instance name of SQL Server is currently installed.

> **Note:** For input parameter descriptions and defaults, see "Parameters for MS SQL - Upgrade Standalone SQL Instance" on page 106.

How to Run this Workflow

The following instructions show you how to customize and run the MS SQL Upgrade Standalone SQL Instance workflow in your environment.

The workflow provides default values for some parameters. These default values are usually sufficient for a "typical" installation. You can override the defaults by specifying parameter values in the deployment. You can also expose additional parameters in the workflow, if necessary, to accomplish more advanced scenarios. Any parameters not explicitly specified in the deployment will have the default values listed in "Parameters for MS SQL - Upgrade Standalone SQL Instance" on page 106.

> **Note:** Before following this procedure, review the "Prerequisites for this Workflow" on page 101, and ensure that all requirements are satisfied.

To use the Run MS SQL - Upgrade Standalone SQL Instance workflow:

1. Create a deployable copy of the workflow.
2. Determine the values that you will specify for the following parameters:

   > **Note:** There are no mandatory parameters required to run this workflow. All parameters are optional. You may need to expose additional parameters depending on your objectives.
   >
   > See "Parameters for MS SQL - Upgrade Standalone SQL Instance" on page 106 for detailed descriptions of all input parameters for this workflow, including default values.

3. In the workflow editor, expose any additional parameters that you need. You will specify values for those parameters when you create the deployment or at runtime.
4. Save the changes to the workflow (click **Save** in the lower right corner).
5. Create a new deployment.
6. On the Parameters tab, specify values (or set the type to Runtime Value) for the required parameters listed in step 2 and any additional parameters that you have exposed. You do not need to specify values for those parameters whose default values are appropriate for your environment.
7. On the Targets tab, specify one or more targets for this deployment.
8. Save the deployment (click **Save** in the lower right corner).
9. Run the workflow using this deployment, specifying any runtime parameters.

To verify the results:

The workflow will complete and report SUCCESS on the Console if it has run successfully. If an error occurs during workflow execution, the error is logged, and the workflow terminates in the FAILURE state. Also verify by checking that the target database no longer appears in the DMA Environment section.

Parameters for MS SQL - Upgrade Standalone SQL Instance

The following tables describe the required and optional input parameters for this workflow. Some of these parameters may not be initially visible in a deployment. For some parameters, if you do not specify a value for a parameter, a default value is assigned.

> **Note:** Only those parameters that are configurable in a standard deployment are listed here. Input parameters that must be mapped to output parameters of previous steps are not listed.

Parameters Defined in this Step: MS SQL - Parameters - Upgrade Standalone

Parameter Name	Default Value	Required	Description
Download From Software Directory	no default	optional	The name of the ZIP file containing the SQL Server setup files
Download Target Destination	no default	required	The local directory where the SQL Setup files should be stored.
Instance Name	MSSQLSERVER	required	The name of the newly created instance. Use MSSQLSERVER for the default instance, any other alphanumeric value for a named instance.

Additional Parameters Defined in this Step: MS SQL - Advanced Parameters - Upgrade Standalone

Parameter Name	Default Value	Required	Description
Additional Template Parameters	no default	optional	Pipe-delimited ("|") list of additional template parameters and values. `SQMREPORTING|1|INSTANCEDIR|"D:\SQLDirectory`
Installation Path	no default	optional	Specifies the location for the SQL Server program files.
Installer Account	no default	optional	The Windows account that will be performing the installation.
Installer Password	no default	optional	The password of the Windows account that will be performing the installation.

Additional Parameters Defined in this Step: MS SQL - Advanced Parameters - Upgrade Standalone, continued

Parameter Name	Default Value	Required	Description
Product Key	no default	optional	Specifies the product key for the edition of SQL Server. If this parameter is not specified, Evaluation is used.
Skip Simulation	no default	optional	If set to "YES", workflow will skip Simulate step and proceed directly to install/upgrade step.

MS SQL Rollback Patch

This workflow is supported on the Windows operating system platform. The MS SQL Rollback Patch enables you to uninstall a SQL Server patch on a standalone 2005/2008/2008R2 instance.

To use this workflow in your environment, see the following information:

Topic	Information Included
"Prerequisites for this Workflow" on the next page	List of prerequisites that must be satisfied before you can run this workflow
"How this Workflow Works" on page 110	Information about what the workflow does, including validation checks performed, steps executed, and step descriptions
"How to Run this Workflow" on page 112	Instructions for running this workflow in your environment
"Parameters for MS SQL Rollback Patch" on page 114	List of input parameters for this workflow

Prerequisites for this Workflow

Be sure that the following prerequisites are satisfied before you run the MS SQL Rollback Patch workflow:

- This solution requires HPE DMA version 10.30 (or later).

  The latest HPE DMA solution packs require the latest HPE DMA platform. To use the latest solution packs, update the HPE DMA platform. HPE DMA10.50 solution packs are supported on HPE DMA10.50 (and later).
- You have installed the Database Compliance solution pack.

The workflow must be able to:

- MS SQL service should be up and running.
- Log in to the MS SQL instance using MS SQL login credentials.
- It should drop the database upon connecting to the MS SQL instance.

The information presented here assumes the following: show assumptions

- HPE DMA is installed and operational.
- At least one suitable target server is available.
- You are logged in to the HPE DMA web interface.
- You have permission to create, edit, and deploy copies of the workflows included in this solution pack.

For more information about prerequisites for MS SQL database, refer to the MS SQL Server Documentation.

How this Workflow Works

This workflow performs the following actions:

Uninstalls a SQL Server patch on a standalone 2005/2008/2008R2 instance. The default deployment will only show required parameters.

Steps Executed by the Workflow

The MS SQL Rollback Patch workflow includes the following steps. Each step must complete successfully before the next step can start. If a step fails, the workflow reports a failure and all subsequent steps are skipped.

Steps Used by MS SQL Rollback Patch

Workflow Step	Description
MS SQL Parameters Rollback Patch	This step gathers all the required parameters for a rollback (uninstall) of a SQL Server patch.
MS SQL Gather Advanced Parameters for Rollback Patch	This step gathers all the advanced parameters for a rollback (uninstall) of a SQL Server patch.
Windows Check for Pending Reboot	This step check for any pending reboots.
Check If Download File Exists	This step is designed to facilitate the complicated methodologies that various companies use to distribute their software bundles for installation.
MS SQL Verify Patch Rollback	This step verifies that a rollback of a Windows or SQL Server patch was successful.
Delete Directory	This step deletes a directory (folder).
Windows Restart Server	This step Restart a system
Download Software	This step automates the transfer of files from the HP SA Software Library to individual managed servers for use in downstream workflow steps.
MS SQL Rollback Patch	This step performs a rollback on a Windows or SQL Server patch.
Windows Wait for Restart	This step is to wait 8 minutes for Windows server to finish restart.
Unzip Archive	This step is to unzip a zip archive, verify if the input file exists, ensure the output directory exists, creates required directories, and deploys archived files.
MS SQL Verify Patch Rollback	This step verifies that a rollback of a Windows or SQL Server patch was successful.

Steps Used by MS SQL Rollback Patch, continued

Workflow Step	Description
Delete File	This step deletes the specified file.
Windows Check for Pending Reboot	This step checks for any pending reboots.
Delete Directory	This step deletes a directory (folder).
Windows Restart Server	This step restarts a system.
Discover SQL Databases	This step audits the server's physical environment looking for SQLServer instances and databases.
Windows Wait for Restart	This step is to wait 8 minutes for Windows server to finish restart.

Note: For input parameter descriptions and defaults, see "Parameters for MS SQL Rollback Patch" on page 114.

How to Run this Workflow

The following instructions show you how to customize and run the MS SQL Rollback Patch workflow in your environment.

The workflow provides default values for some parameters. These default values are usually sufficient for a "typical" installation. You can override the defaults by specifying parameter values in the deployment. You can also expose additional parameters in the workflow, if necessary, to accomplish more advanced scenarios. Any parameters not explicitly specified in the deployment will have the default values listed in "Parameters for MS SQL Rollback Patch" on page 114.

> **Note:** Before following this procedure, review the "Prerequisites for this Workflow" on page 109, and ensure that all requirements are satisfied.

To use the Run MS SQL Rollback Patch workflow:

1. Create a deployable copy of the workflow.
2. Determine the values that you will specify for the following parameters:

   > **Note:** There are no mandatory parameters required to run this workflow. All parameters are optional. You may need to expose additional parameters depending on your objectives.
   >
   > See "Parameters for MS SQL Rollback Patch" on page 114 for detailed descriptions of all input parameters for this workflow, including default values.

3. In the workflow editor, expose any additional parameters that you need. You will specify values for those parameters when you create the deployment or at runtime.
4. Save the changes to the workflow (click **Save** in the lower right corner).
5. Create a new deployment.
6. On the Parameters tab, specify values (or set the type to Runtime Value) for the required parameters listed in step 2 and any additional parameters that you have exposed. You do not need to specify values for those parameters whose default values are appropriate for your environment.
7. On the Targets tab, specify one or more targets for this deployment.
8. Save the deployment (click **Save** in the lower right corner).
9. Run the workflow using this deployment, specifying any runtime parameters.

To verify the results:

The workflow will complete and report SUCCESS on the Console if it has run successfully. If an error occurs during workflow execution, the error is logged, and the workflow terminates in the FAILURE state. Also verify by checking that the target database no longer appears in the DMA Environment section.

Parameters for MS SQL Rollback Patch

The following tables describe the required and optional input parameters for this workflow. Some of these parameters may not be initially visible in a deployment. For some parameters, if you do not specify a value for a parameter, a default value is assigned.

> **Note:** Only those parameters that are configurable in a standard deployment are listed here. Input parameters that must be mapped to output parameters of previous steps are not listed.

Input Parameters Defined in this Step: MS SQL Parameters Rollback Patch

Parameter Name	Default Value	Required	Description
Patch Name	no default	required	Name of the patch, the KB number of the patch, or "Latest Patch" to automatically rollback latest patch on instance. This field is case-insensitive.

MS SQL - Create AlwaysOn Availability Group

The MS SQL - Create AlwaysOn Availability Group workflow creates a new AlwaysOn Availability Group on the primary target, then adds secondary replicas to the group. Member databases will then be added to the Availability Group, while replica configuration is handled during the entire process.

To use this workflow in your environment, see the following information:

Topic	Information Included
"Prerequisites for this Workflow" on the next page	List of prerequisites that must be satisfied before you can run this workflow
"How this Workflow Works" on page 117	Information about what the workflow does, including validation checks performed, steps executed, and step descriptions
" How to Run this Workflow" on page 119	Instructions for running this workflow in your environment
"Parameters for MSSQL - Create AlwaysOn Availability Group" on page 121	List of input parameters for this workflow

Prerequisites for this Workflow

Be sure that the following prerequisites are satisfied before you run the MS SQL - Create AlwaysOn Availability Group workflow:

- This solution requires HPE DMA version 10.50 (or later).

  The latest HPE DMA solution packs require the latest HPE DMA platform. To use the latest solution packs, update the HPE DMA platform. HPE DMA10.50 solution packs are supported on HPE DMA10.50 (and later).
- You have installed the Database Provisioning solution pack.
- An existing SQL server instance to be used as the target instance.
- Workflow needs to run against nodes that are members of the same Windows cluster.
- Each workflow target should be a standalone instance that is installed on a cluster node.
- Workflow should run under a domain account that has access to all instances to be added to new Availability Group, as well as has access to the Windows share where backup files will be saved.

The information presented here assumes the following: show assumptions

- HPE DMA is installed and operational.
- At least one suitable target server (database) is available.
- You are logged in to the HPE DMA web interface.
- You have permission to create, edit, and deploy copies of the workflows included in this solution pack.

For more information about prerequisites for MySQL database, refer to the Microsoft SQL Server Documentation.

How this Workflow Works

This workflow performs the following actions:

- Creates a new AlwaysOn Availability Group on the primary target, then adds secondary replicas to the group.

Steps Executed by the Workflow

The MS SQL - Create AlwaysOn Availability Group workflow includes the following steps. Each step must complete successfully before the next step can start. If a step fails, the workflow reports a failure and all subsequent steps are skipped.

Steps Used by MS SQL - Create AlwaysOn Availability Group

Workflow Step	Description
MS SQL - Gather Parameters for AlwaysOn Group	This step gathers parameters to create AlwaysOn availability group.
MS SQL - Gather Advanced Parameters for AlwaysOn Group	This steps gathers advanced parameters to create AlwaysOn availability group.
MS SQL - Check AlwaysOn Prerequisites	This step checks for pre-requisites that are mandatory to create AlwaysOn group if the Windows version is greater than 2008, the installed SQL server is an Enterprise edition, and the server that if the AlwaysOn group is not a domain controller.
MS SQL - Enable AlwaysOn	This step enables the AlwaysOn feature on the instance that will be added to the AlwaysOn group.
MS SQL - Create Mirroring Endpoint	This step creates the endpoint and grants connect permission to the created endpoint.
MS SQL - Run Setup AlwaysOn Secondary	This step triggers the execution of subflow MS SQL - Setup AlwaysOn Secondary on the secondary servers.
MS SQL - Backup Databases for AlwaysOn	This step creates backup databases on an instance (Full, Differential, or Log backup types). The list of databases to backup can range from all databases (default), all except a select few (ignore list), or just a select few (exclusive list).
MS SQL - Create AlwaysOn Group	This step creates the AlwaysOn group.
MS SQL - Backup Databases for AlwaysOn	This step creates backup databases on an instance (Full, Differential, or Log backup types). The list of databases to backup can range from all databases (default), all except a select few (ignore list), or just a select few (exclusive list).

Steps Used by MS SQL - Create AlwaysOn Availability Group, continued

Workflow Step	Description
MS SQL - Run Join Secondary to AlwaysOn Group	This step triggers the subflow MS SQL - Join Secondary to AlwaysOn Group that in turn adds the secondary server to the AlwaysOn group.
MS SQL - Validate AlwaysOn Availability Group	This step validates, if the AlwaysOn group has been created appropriately.

How to Run this Workflow

The following instructions show you how to customize and run the MS SQL - Create AlwaysOn Availability Group workflow in your environment.

The workflow provides default values for some parameters. These default values are usually sufficient for a "typical" installation. You can override the defaults by specifying parameter values in the deployment. You can also expose additional parameters in the workflow, if necessary, to accomplish more advanced scenarios. Any parameters not explicitly specified in the deployment will have the default values listed in "Parameters for MSSQL - Create AlwaysOn Availability Group" on page 121.

> **Note:** Before following this procedure, review the "Prerequisites for this Workflow" on page 116, and ensure that all requirements are satisfied.

To use the MS SQL - Create AlwaysOn Availability Group workflow:

1. Create a deployable copy of the workflow.
2. Determine the values that you will specify for the parameters.

   > **Note:** There are no mandatory parameters required to run this workflow. All parameters are optional. You may need to expose additional parameters depending on your objectives.

3. In the workflow editor, expose any additional parameters that you need. You will specify values for those parameters when you create the deployment or at runtime.
4. Save the changes to the workflow (click **Save** in the lower right corner).
5. Create a new deployment.
6. On the Parameters tab, specify values (or set the type to Run time Value) for the required parameters listed in step 2 and any additional parameters that you have exposed. You do not need to specify values for those parameters whose default values are appropriate for your environment.
7. On the Targets tab, specify one or more targets for this deployment.
8. Save the deployment (click **Save** in the lower right corner).
9. Run the workflow using this deployment, specifying any runtime parameters.

To verify the results:

The workflow will complete and report SUCCESS on the Console if it has run successfully. If an error occurs during workflow execution, the error is logged, and the workflow terminates in the FAILURE state. The database will be removed from the DMA environment section upon SUCCESS as well.

Use SQL Server Management Studio to verify that Availability Group has been created (see http://msdn.microsoft.com/en-us/library/ff878267.aspx for more information).

.

Parameters for MSSQL - Create AlwaysOn Availability Group

The following tables describe the required and optional input parameters for this workflow. Some of these parameters are not initially visible in a deployment. For most parameters, if you do not specify a value for a parameter, a default value is assigned.

Parameters Defined in this Step: MS SQL - Gather Parameters for AlwaysOn Group

Parameter Name	Default Value	Required	Description
Databases in Group	no default	required	Comma-separated list of database names to be included in Availability Group.
List of Secondary Server-Instances	no default	required	Comma-separated list of server-instances to be secondaries. For example: Server1\Instance1,Server2\Instance2,Server3\Instance3.
Mirroring Endpoint Port	4022	required	Specifies the port number listened to for connections by the service broker TCP/IP protocol. Default is 4022. Valid values are between 1024 and 32767.
Path to Share for Backup Files	no default	required	A Windows share location that all the cluster nodes can access, which will store backup files for the group databases.
Primary Availability Mode	SYNCHRONOUS	required	Specifies whether the primary replica has to wait for the secondary replica to acknowledge the hardening (writing) of the log records to disk before the primary replica can commit the transaction on a given primary database. Valid values are SYNCHRONOUS and ASYNCHRONOUS.
Primary Failover Mode	AUTOMATIC	required	Specifies the failover mode of the primary instance. Valid values are AUTOMATIC and MANUAL.
Secondary Availability Modes	SYNCHRONOUS	required	Comma-separated list of availability modes of secondary instances. Including the primary instance, you can specify up to three instances with SYNCHRONOUS mode, while up to five can be specified with ASYNCHRONOUS mode.
Secondary Failover Modes	AUTOMATIC	required	Comma-separated list of Failover Modes of secondary instances. Including the primary instance, you can specify up to two instances with AUTOMATIC mode,

Parameters Defined in this Step: MS SQL - Gather Parameters for AlwaysOn Group, continued

Parameter Name	Default Value	Required	Description
			while up there is no limit on instances with MANUAL mode.

Parameters Defined in this Step: MS SQL - Gather Advanced Parameters for AlwaysOn Group

Parameter Name	Default Value	Required	Description
Instance Password	no default	optional	Password for the instance that will be added to AlwaysOn group.
Instance User	no default	optional	User account to access the instance that will be added to AlwaysOn group.
List of Server-Instances	no default	optional	Comma-separated list of server-instances to be secondaries. For example: Server1\Instance1,Server2\Instance2,Server3\Instance3.
Primary Port Number	4022	optional	Specifies the port number listened to for connections by the service broker TCP/IP protocol. Default is 4022. Valid values are between 1024 and 32767.
Secondary Port Numbers	no default	optional	Comma-separated list of port numbers that will be used on the secondary server.
Subflow Parallel Execution	yes	optional	Value to represent whether all the secondary can be joined to the primary in parallel. Default is yes. Valid values are true, false, yes, and no.
Web API - Password	no default	optional	DMA user account password.
Web API - URL	no default	optional	DMA server URL.
Web API - Username	no default	optional	DMA user account.

MS SQL - Install Clustered SQL Instance

This workflow installs a new standalone instance of SQL Server 2008/2008 R2/2012/2014 on an already existing Windows 2003/2008/2008 R2/2012/2012 R2 server. The default deployment will only show required parameters.

To use this workflow in your environment, see the following information:

Topic	Information Included
"Prerequisites" on page 137	List of prerequisites that must be satisfied before you can run this workflow
"How this workflow works" on page 138	Information about what the workflow does, including validation checks performed, steps executed, and step descriptions
"How to run this workflow" on page 139	Instructions for running this workflow in your environment
"Parameters for MS SQL - Add Node to Cluster" on page 144	List of input parameters for this workflow

Prerequisites

Before performing the procedures in this section, your environment must meet the following minimum requirements:

- An existing Windows 2008, 2008 R2, or 2012 cluster
- Installation software:

  The SQL Server 2008, 2008 R2, or 2012 software installation files, obtained from Microsoft.

  The installation media must be available locally or available for download from the software repository.
- Storage:

  An available shared disk for SQL Server shared files

  A staging directory with 4 gigabytes available to unzip the SQL Server software
- Permissions to create an SQL Server database:

  System Stored Procedures (SP)

  CREATE LOGIN

  If using a non-default database owner, the `sp_changedbowner` process is available

  If a non-default database owner is specified and does not exist, permission to create the appropriate login
- .NET 3.5 is installed.

> **Note:** For additional information, see "Run as a Windows Domain User" in the *HPE DMA Installation Guide*, available at: https://softwaresupport.hp.com/

- Licenses for SQL Server and HPE DMA.

For additional requirements, see the following Microsoft documentation:

SQL Server version	Microsoft documentation
2008	Hardware and Software Requirements for Installing SQL Server 2008
2008 R2	Hardware and Software Requirements for Installing SQL Server 2008 R2
2012	Hardware and Software Requirements for Installing SQL Server 2012

How this workflow works

This workflow performs the following actions:

Installs a new clustered instance of SQL Server 2008, 2008 R2, 2012, or 2014 on an already existing Windows 2008/2008 R2/2012/2012 R2 cluster.

Steps Executed

The MS SQL - Install Clustered SQL Instance workflow includes the following steps. Each step must complete successfully before the next step can start. If a step fails, the workflow reports a failure and all subsequent steps are skipped.

Steps used by MS SQL - Install Clustered SQL Instance

Workflow Step	Description
MS SQL - Gather Parameters For Install Clustered SQL Instance	This step gathers all the required parameters for a clustered SQL 2008 install.
MS SQL - Gather Advanced Parameters For Install Clustered SQL Instance	This step gathers all the optional parameters for a clustered SQL 2008 install.
Check If Download File Exists	This step is designed to facilitate the complicated methodologies that various companies use to distribute their software bundles for installation.
MS SQL - Create Install or Upgrade Template	This step verifies that all required parameters are provided, and writes any optional parameters to the template file if they are non-blank.

Steps used by MS SQL - Install Clustered SQL Instance, continued

Workflow Step	Description
Download Software	This step automates the transfer of files from the HP SA Software Library to individual managed servers for use in downstream workflow steps. Verifies checksum of each file transferred.
Unzip Archive	This step unzips a "zip" archive, verifies that the input file exists, ensures that output directory exists, creates required directories, and deploys archived files.
Delete File	This step verifies a specified file exists and deletes it.
MS SQL - Simulate - Install or Upgrade	This step verifies that all required parameters are provided, and the system meets minimum requirements.
Delete File	This step verifies a specified file exists and deletes it.
MS SQL - Install or Upgrade	This step installs SQL Server 2008 by running the setup.exe program located on the installation media.
MS SQL Verify SQL Installation	This step determines if the target instance name of SQL Server is currently installed.
Delete Directory	This directory verifies a specified file exists and deletes it.
Delete File	This step verifies a specified file exists and deletes it.
Windows Check for Pending Reboot	Check for any pending reboots. This ensures that an installation can be run without a prior reboot requirement.
Discover SQL Databases	Audits the server's physical environment looking for SQLServer instances and databases.
Windows Restart Server	Restarts a system Input Wait Time: The number of seconds to wait before the reboot.
Delete File	This step verifies a specified file exists and deletes it.
Windows Check for Pending Reboot	Checks for any pending reboots. This ensures that an installation can be run without a prior reboot requirement.
Windows Wait for Restart	Waits 8 minutes for Windows server to finish restart.
Windows Restart Server	Restarts a system Input Wait Time: The number of seconds to wait before the reboot.
MS SQL - Install or Upgrade	This step installs SQL Server 2008 by running the setup.exe program located on the installation media.
Windows Wait for Restart	Waits 8 minutes for Windows server to finish restart.
MS SQL Verify SQL Installation	This step determines if the target instance name of SQL Server is currently installed.

How to run this workflow

The following instructions show you how to customize and run the MS SQL - Install Standalone SQL Instance workflow in your environment.

The workflow provides default values for some parameters. These default values are usually sufficient for a "typical" installation. You can override the defaults by specifying parameter values in the deployment. You can also expose additional parameters in the workflow, if necessary, to accomplish more advanced scenarios. Any parameters not explicitly specified in the deployment will have the default values listed in Parameters for MS SQL - Install Standalone SQL Instance.

> **Note:** Before following this procedure, review the Prerequisites, and ensure that all requirements are satisfied.

To use the MS SQL - Install Standalone SQL Instance workflow:

1. Create a deployable copy of the workflow (see "Create a Deployable Workflow" in *HPE DMA Quick Start Tutorial*)
   a. Determine the values that you will specify for the following parameters.

      The following tables describe the required and optional input parameters for this workflow.

      Step: MS SQL - Parameters - Install Standalone

Parameter	Description	Example Value
Download From Software Directory	Optional: The name of the ZIP file that contains the SQL Server installation software files obtained from Microsoft.[1] **Note:** If necessary, manually zip the installation software files up.	`SQL12.zip`
Download Target Destination	Required: The local directory where the SQL Server setup files are stored:[2] If source files are in the software repository: Location where	`C:\temp`

[1] If the file is not found on the target server(s), it will be downloaded from the software repository. For additional information, see Alternative methods for specifying input files.
[2] For additional information, see Alternative methods for specifying input files.

Step: MS SQL - Parameters - Install Standalone, continued

Parameter	Description	Example Value
	Download From Software Directory will be downloaded and extracted If source files are on the target: Location where the Microsoft SQL Server installation files already exist—not zipped up Upon a successful workflow completion, all downloaded and extracted files are cleaned up.	
Instance Name	Required: The name of the newly created virtual server and instance. Format: <Virtual Server>\<Instance Name> Use MSSQLSERVER for the default instance and any other alphanumeric value for a named instance.	SQL-CLUSTER\InstanceA

Step: MS SQL - Advanced Parameters - Install Standalone

Parameter	Description	Example Value
Additional Template Parameters	Optional: Pipe-delimited (|) list of additional template parameters and values. Should follow this example: SQMREPORTING|1| INSTANCEDIR|"D:\SQLDirectory"	SQMREPORTING|1| INSTANCEDIR|"D:\SQLDirectory"
Data File Location	Optional: The location for the SQL Server program files.	?
Install Components	Optional: A comma-delimited list that specifies which components to install. Feature names are case sensitive. For a list of components for SQL Server 2008 R2 (as well as links to previous versions), see: msdn.microsoft.com/en-us/library/ms144259(v=SQL.105).aspx#Feature	?

Step: MS SQL - Advanced Parameters - Install Standalone, continued

Parameter	Description	Example Value
Installation Path	Optional: The location for the SQL Server program files.	?
Installer Account	Optional: The Windows account that will perform the installation.	?
Installer Password	Optional: The password of the Windows account that will perform the installation	?
Product Key	Optional: Specifies the product key for the edition of SQL Server. If this parameter is not specified, Evaluation is used.	?
SA Password	Optional: The password for the SQL Server SA account. If specified, the security mode will be set to SQL authentication. If left blank, security mode will be set to Windows authentication.	?
SQL Agent Account	Optional: The login account for the SQL Server Agent service. Can be a local Windows user, a domain user, or a built-in account (for example, NT AUTHORITY\NETWORK SERVICE). If not a built-in account, also specify SQL Agent Password.	Win12\Administrator
SQL Agent Password	Optional: Specify if SQL Agent Account is specified.	●●●
SQL Browser Account	Optional: The login account for the SQL Server Agent service. Can be a local Windows user, a domain user, or a built-in account (for example, NT AUTHORITY\NETWORK SERVICE). If not a built-in account, also specify SQL Browser Password.	?
SQL Browser	Required if SQL Browser Account	?

Step: MS SQL - Advanced Parameters - Install Standalone, continued

Parameter	Description	Example Value
Password	is specified and is not a built-in account.	
SQL Cluster Domain Group	Optional: The domain group that the SQL Service Account user is a member of.	?
SQL Cluster Resource Group	Optional: The name of the cluster resource group where the SQL cluster will be installed. This cluster group should already be created by a system administrator. The cluster resource group will have the shared disks where the SQL data files and shared program files will be stored.	?
SQL Collation	Optional: The collation of the instance. If left blank, the instance will be installed with the collation of the OS.	?
SQL Service Account	Optional: The login account for the SQL Server service. Can be a local Windows user, a domain user, or a built-in account (for example, NT AUTHORITY\NETWORK SERVICE). If not a built-in account, also specify SQL Service Password.	Win12\Administrator
SQL Service Password	Required if SQL Service Account is specified and is not a built-in account.	●●●
SQL Sysadmin Accounts	Optional, only applies to SQL Server 2008 and 2008 R2 installs. Not applicable for SQL Server 2005. A comma-delimited list of user accounts that will be set as system administrators. Each account must either be a local Windows user or a domain user. Required: A comma-delimited list of user accounts that will be set as	Win12\Administrator

Step: MS SQL - Advanced Parameters - Install Standalone, continued

Parameter	Description	Example Value
	system administrators. Each account must either be a local Windows user or a domain user. This parameter is optional for SQL Server 2008 or 2008 R2.	
Skip Simulation	If set to "YES", workflow will skip Simulate step and proceed directly to install/upgrade step	NA

2. In the workflow editor, expose any additional parameters that you need. You will specify values for those parameters when you create the deployment or at runtime.
3. Save the changes to the workflow (click **Save** in the lower right corner).
4. Create a new deployment. See "Create a Deployment" in *HPE DMA Quick Start Tutorial* for instructions.
5. On the Parameters tab, specify values (or set the type to Runtime Value) for the required parameters listed in step 2 and any additional parameters that you have exposed. You do not need to specify values for those parameters whose default values are appropriate for your environment.
6. On the Targets tab, specify one or more targets for this deployment.
7. Save the deployment (click **Save** in the lower right corner).
8. Run the workflow using this deployment, specifying any runtime parameters. See "Run Your Workflow" in (*HPE DMA Quick Start Tutorial* for instructions.

To verify the results:

The workflow will complete and report SUCCESS on the Console if it has run successfully. If an error occurs during workflow execution, the error is logged, and the workflow terminates in the FAILURE state.

Parameters for MS SQL - Install Clustered SQL Instance

The following tables describe the required and optional input parameters for this workflow.

Step: MS SQL - Gather Parameters For Install Clustered SQL Instance

Parameter	Required	Example Value	Description
Cluster Administrator Account	required	Win12\Administrator	The Windows domain user that will run the setup operation. This user requires elevated administrator privileges on the cluster. Format: <DOMAIN>\<USERNAME>
Cluster Administrator Password	required	●●●	Password for the Windows domain user that will run the setup operation. This user requires elevated administrator privileges on the cluster. Must be a strong Windows password.
Download From Software Directory	optional	`SQL12.zip`	The name of the ZIP file that contains the SQL Server installation software files obtained from Microsoft.[1] **Note:** If necessary, manually zip the installation software files up.
Download Target Destination	required	`C:\temp`	The local directory where the SQL Server setup files are stored:[2] If source files are in the software repository: Location where Download From Software Directory will be downloaded and extracted If source files are on the target: Location where the Microsoft SQL Server installation files already exist—not zipped up Upon a successful workflow completion, all downloaded and extracted files are cleaned up.
Instance Name	required	SQL-CLUSTER\InstanceA	The name of the newly created virtual server and instance. Format: <Virtual Server>\<Instance Name> Use MSSQLSERVER for the default

[1]If the file is not found on the target server(s), it will be downloaded from the software repository. For additional information, see Alternative methods for specifying input files.
[2]For additional information, see Alternative methods for specifying input files.

Step: MS SQL - Gather Parameters For Install Clustered SQL Instance, continued

Parameter	Required	Example Value	Description
			instance and any other alphanumeric value for a named instance.
Public IP Address	required	DHCP	Public IP Address. For SQL Server 2012 set to DHCP.
Public IP Network Name	required		IP Network Name for the clusters. Format: <Network Name>:<Subnet Mask> For example: Public:255.255.255.0
SQL Agent Account	required	Win12\Administrator	The login account for the SQL Server Agent service. Can be a local Windows user, a domain user, or a built-in account (for example: NT AUTHORITY\NETWORK SERVICE). If not a built-in account, also specify SQL Agent Password. This parameter is optional for SQL Server 2008 or 2008 R2.
SQL Agent Password	required	●●●	Specify if SQL Agent Account is specified. This parameter is optional for SQL Server 2008 or 2008 R2.
SQL Service Account	required	Win12\Administrator	The login account for the SQL service. Can be a local Windows user, a domain user, or a built-in account (for example: NT AUTHORITY\NETWORK SERVICE). If not a built-in account, also specify SQL Service Password. This parameter is optional for SQL Server 2008 or 2008 R2.
SQL Service Password	required	●●●	Specify if SQL Service Account is specified. This parameter is optional for SQL Server 2008 or 2008 R2.
SQL Sysadmin Accounts	required	Win12\Administrator	A comma-delimited list of user accounts that will be set as system

Step: MS SQL - Gather Parameters For Install Clustered SQL Instance, continued

Parameter	Required	Example Value	Description
			administrators. Each account must either be a local Windows user or a domain user. This parameter is optional for SQL Server 2008 or 2008 R2.

Step: MS SQL - Gather Advanced Parameters For Install Clustered SQL Instance

Parameter	Required	Example Value	Description
Additional Template Parameters	optional	SQMREPORTING|1| INSTANCEDIR|"D:\SQLDirectory"	Pipe-delimited (|) list of additional template parameters and values. Should follow this example: SQMREPORTING|1| INSTANCEDIR|"D:\SQLDirectory"
Cluster Node Names	optional	?	Comma-delimited list of node members' hostnames, including target hostname. Acceptable format: [Node1Hostname], [Node2Hostname]
Data File Location	optional	?	The location for the SQL Server program files.
Install Components	optional	?	A comma-delimited list that specifies which components to install. Feature names are case sensitive. For a list of components for SQL Server 2008 R2 (as well as links to previous versions), see: msdn.microsoft.com/en-us/library/ms144259(v=SQL.105).aspx#Feature
Installation Path	optional	?	The location for the SQL Server program files.
Installer Account	optional	?	The Windows account that will perform the installation.
Installer Password	optional	?	The password of the Windows account that will perform the installation

Step: MS SQL - Gather Advanced Parameters For Install Clustered SQL Instance, continued

Parameter	Required	Example Value	Description
Product Key	optional	?	Specifies the product key for the edition of SQL Server. If this parameter is not specified, Evaluation is used.
SA Password	optional	?	The password for the SQL Server SA account. If specified, the security mode will be set to SQL authentication. If left blank, security mode will be set to Windows authentication.
SQL Agent Account	optional	Win12\Administrator	The login account for the SQL Server Agent service. Can be a local Windows user, a domain user, or a built-in account (for example, NT AUTHORITY\NETWORK SERVICE). If not a built-in account, also specify SQL Agent Password.
SQL Agent Domain Group	optional	?	The domain group that the SQL Agent Account user is a member of.
SQL Agent Password	optional	●●●	Specify if SQL Agent Account is specified.
SQL Browser Account	optional	?	The login account for the SQL Server Agent service. Can be a local Windows user, a domain user, or a built-in account (for example, NT AUTHORITY\NETWORK SERVICE). If not a built-in account, also specify SQL Browser Password.
SQL Browser Password	required	?	Required if SQL Browser Account is specified and is not a built-in account.
SQL Cluster Domain Group	optional	?	The domain group that the SQL Service Account user is a member of.
SQL Cluster	optional	?	The name of the cluster resource

Step: MS SQL - Gather Advanced Parameters For Install Clustered SQL Instance, continued

Parameter	Required	Example Value	Description
Resource Group			group where the SQL cluster will be installed. This cluster group should already be created by a system administrator. The cluster resource group will have the shared disks where the SQL data files and shared program files will be stored.
SQL Cluster Shared Directory	optional	?	The path to the directory where the shared cluster program files will be stored. Must be on a disk shared by all nodes of the cluster.
SQL Collation	optional	?	The collation of the instance. If left blank, the instance will be installed with the collation of the OS.
SQL Service Account	optional	Win12\Administrator	The login account for the SQL Server service. Can be a local Windows user, a domain user, or a built-in account (for example, NT AUTHORITY\NETWORK SERVICE). If not a built-in account, also specify SQL Service Password.
SQL Service Password	required	●●●	Required if SQL Service Account is specified and is not a built-in account.
SQL Sysadmin Accounts	required	Win12\Administrator	Optional, only applies to SQL Server 2008 and 2008 R2 installs. Not applicable for SQL Server 2005. A comma-delimited list of user accounts that will be set as system administrators. Each account must either be a local Windows user or a domain user. A comma-delimited list of user accounts that will be set as system administrators. Each account must either be a local Windows user or a domain user. This parameter is optional for

Step: MS SQL - Gather Advanced Parameters For Install Clustered SQL Instance, continued

Parameter	Required	Example Value	Description
			SQL Server 2008 or 2008 R2.
TempDB Data Directory	NA	NA	Specifies the directory for the data files for tempdb.
TempDB Log Directory	NA	NA	Specifies the directory for the log files for tempdb.
Update Source	NA	MU	The location where SQL Server setup will obtain product updates. The valid values are "MU" to search Microsoft Update, a valid folder path, a relative path such as .\MyUpdates or a UNC share.

MS SQL - Add Node to Cluster v2

This workflow installs a new clustered instance of SQL Server 2008, 2008 R2, 2012, or 2014 on an already existing Windows 2008/2008 R2/2012/2012 R2 cluster. The default deployment will only show required parameters.

To use this workflow in your environment, see the following information:

Topic	Information Included
"Prerequisites" on the next page	List of prerequisites that must be satisfied before you can run this workflow
"How this workflow works" on page 138	Information about what the workflow does, including validation checks performed, steps executed, and step descriptions
"How to run this workflow" on page 139	Instructions for running this workflow in your environment
"Parameters for MS SQL - Add Node to Cluster" on page 144	List of input parameters for this workflow

Prerequisites

Before performing the procedures in this section, your environment must meet the following minimum requirements:

- An existing Windows 2008, 2008 R2, or 2012 cluster
- Installation software:

  The SQL Server 2008, 2008 R2, or 2012 software installation files, obtained from Microsoft.

  The installation media must be available locally or available for download from the software repository.
- Storage:

  An available shared disk for SQL Server shared files

  A staging directory with 4 gigabytes available to unzip the SQL Server software
- Permissions to create an SQL Server database:

  System Stored Procedures (SP)

  CREATE LOGIN

  If using a non-default database owner, the `sp_changedbowner` process is available

  If a non-default database owner is specified and does not exist, permission to create the appropriate login
- .NET 3.5 is installed.

  > **Note:** For additional information, see "Run as a Windows Domain User" in the *HPE DMA Installation Guide*, available at: https://softwaresupport.hp.com/
- Licenses for SQL Server and HPE DMA.

For additional requirements, see the following Microsoft documentation:

SQL Server version	Microsoft documentation
2008	Hardware and Software Requirements for Installing SQL Server 2008
2008 R2	Hardware and Software Requirements for Installing SQL Server 2008 R2
2012	Hardware and Software Requirements for Installing SQL Server 2012

How this workflow works

This workflow installs a new a new standalone instance of SQL Server 2008/2008 R2/2012/2014 on an already existing Windows 2003/2008/2008 R2/2012/2012 R2 server.

Steps Executed

The MS SQL - Add Node to Cluster workflow includes the following steps. Each step must complete successfully before the next step can start. If a step fails, the workflow reports a failure and all subsequent steps are skipped.

Steps used by MS SQL - Add Node to Cluster

Workflow Step	Description
MS SQL - Parameters - Add Node to Cluster	Gathers all the required parameters for a standalone SQL Server install.
MS SQL - Advanced Parameters - Add Node to Cluster V2	Gather all the optional parameters for a standalone SQL Server install
Check If Download File Exists	This step is designed to facilitate the complicated methodologies that various companies use to distribute their software bundles for installation.
MS SQL - Create Install or Upgrade Template	This step verifies that all required parameters are provided, and writes any optional parameters to the template file if they are non-blank.
Download Software	This step automates the transfer of files from the HP SA Software Library to individual managed servers for use in downstream workflow steps. Verifies checksum of each file transferred.
Unzip Archive	This step unzips a "zip" archive, verifies that the input file exists, ensures that output directory exists, creates required directories, and deploys archived files.
Delete File	This step verifies a specified file exists and deletes it.
MS SQL - Simulate - Install or Upgrade	This step verifies that all required parameters are provided, and the system meets minimum requirements.
Delete File	This step verifies a specified file exists and deletes it.
MS SQL - Install or Upgrade	This step installs SQL Server 2008 by running the setup.exe program located on the installation media.
MS SQL Verify SQL Installation	This step determines if the target instance name of SQL Server is currently installed.
Delete Directory	This directory verifies a specified file exists and deletes it.

Steps used by MS SQL - Add Node to Cluster, continued

Workflow Step	Description
Delete File	This step verifies a specified file exists and deletes it.
Windows Check for Pending Reboot	Check for any pending reboots. This ensures that an installation can be run without a prior reboot requirement.
Discover SQL Databases	Audits the server's physical environment looking for SQLServer instances and databases.
Windows Restart Server	Restarts a system Input Wait Time: The number of seconds to wait before the reboot.
Delete File	This step verifies a specified file exists and deletes it.
Windows Check for Pending Reboot	Checks for any pending reboots. This ensures that an installation can be run without a prior reboot requirement.
Windows Wait for Restart	Waits 8 minutes for Windows server to finish restart.
Windows Restart Server	Restarts a system Input Wait Time: The number of seconds to wait before the reboot.
MS SQL - Install or Upgrade	This step installs SQL Server 2008 by running the setup.exe program located on the installation media.
Windows Wait for Restart	Waits 8 minutes for Windows server to finish restart.
MS SQL Verify SQL Installation	This step determines if the target instance name of SQL Server is currently installed.

How to run this workflow

The following instructions show you how to customize and run the MS SQL - Add Node to Cluster workflow in your environment.

The workflow provides default values for some parameters. These default values are usually sufficient for a "typical" installation. You can override the defaults by specifying parameter values in the deployment. You can also expose additional parameters in the workflow, if necessary, to accomplish more advanced scenarios. Any parameters not explicitly specified in the deployment will have the default values listed in "Parameters for MS SQL - Add Node to Cluster" on page 144.

> **Note:** Before following this procedure, review the "Prerequisites" on page 137, and ensure that all requirements are satisfied.

To use the MS SQL - Add Node to Cluster workflow:

1. Create a deployable copy of the workflow (see "Create a Deployable Workflow" in *HPE DMA Quick Start Tutorial*)
   a. Determine the values that you will specify for the following parameters.

      The following tables describe the required and optional input parameters for this workflow.

      Step: MS SQL - Parameters - Add Node to Cluster

Parameter	Description	Example Value
Cluster Administrator Account	Required: The Windows domain user that will run the setup operation. This user requires elevated administrator privileges on the cluster. Format: <DOMAIN>\<USERNAME>	Win12\Administrator
Cluster Administrator Password	Required: Password for the Windows domain user that will run the setup operation. This user requires elevated administrator privileges on the cluster. Must be a strong Windows password.	●●●
Download From Software Directory	Optional: The name of the ZIP file that contains the SQL Server installation software files obtained from Microsoft.[1] **Note:** If necessary, manually zip the installation software files up.	`SQL12.zip`
Download Target Destination	Required: The local directory where the SQL Server setup files are stored:[2] If source files are in the software repository: Location where Download From Software Directory will be downloaded and extracted	`C:\temp`

[1]If the file is not found on the target server(s), it will be downloaded from the software repository. For additional information, see Alternative methods for specifying input files.
[2]For additional information, see Alternative methods for specifying input files.

Step: MS SQL - Parameters - Add Node to Cluster, continued

Parameter	Description	Example Value
	If source files are on the target: Location where the Microsoft SQL Server installation files already exist—not zipped up Upon a successful workflow completion, all downloaded and extracted files are cleaned up.	
Instance Name	Required: The name of the newly created virtual server and instance. Format: <Virtual Server>\<Instance Name> Use MSSQLSERVER for the default instance and any other alphanumeric value for a named instance.	SQL-CLUSTER\InstanceA

Step: MS SQL - Advanced Parameters - Add Node to Cluster

Parameter	Description	Example Value
Additional Template Parameters	Optional: Pipe-delimited (|) list of additional template parameters and values. Should follow this example: SQMREPORTING|1| INSTANCEDIR|"D:\SQLDirectory"	SQMREPORTING|1| INSTANCEDIR|"D:\SQLDirectory"
Installer Account	Optional: The Windows account that will perform the installation.	?
Installer Password	Optional: The password of the Windows account that will perform the installation	?
Product Key	Optional: Specifies the product key for the edition of SQL Server. If this parameter is not specified, Evaluation is used.	?
SQL Agent Account	Optional: The login account for the SQL Server Agent service. Can be a local Windows user, a domain	Win12\Administrator

Step: MS SQL - Advanced Parameters - Add Node to Cluster, continued

Parameter	Description	Example Value
	user, or a built-in account (for example, NT AUTHORITY\NETWORK SERVICE). If not a built-in account, also specify SQL Agent Password.	
SQL Agent Password	Optional: Specify if SQL Agent Account is specified.	●●●
SQL Service Account	Optional: The login account for the SQL Server service. Can be a local Windows user, a domain user, or a built-in account (for example, NT AUTHORITY\NETWORK SERVICE). If not a built-in account, also specify SQL Service Password.	Win12\Administrator
SQL Service Password	Required if SQL Service Account is specified and is not a built-in account.	●●●
Public IP Address	Required: Public IP Address. For SQL Server 2012 set to DHCP.	DHCP
Public IP Network Name	Required: IP Network Name for the clusters. Format: <Network Name>:<Subnet Mask> For example: Public:255.255.255.0	
SQL Agent Account	Required: The login account for the SQL Server Agent service. Can be a local Windows user, a domain user, or a built-in account (for example: NT AUTHORITY\NETWORK SERVICE). If not a built-in account, also specify SQL Agent Password. This parameter is optional for SQL Server 2008 or 2008 R2.	Win12\Administrator

Step: MS SQL - Advanced Parameters - Add Node to Cluster, continued

Parameter	Description	Example Value
SQL Agent Password	Required: Specify if SQL Agent Account is specified. This parameter is optional for SQL Server 2008 or 2008 R2.	●●●
SQL Service Account	Required: The login account for the SQL service. Can be a local Windows user, a domain user, or a built-in account (for example: NT AUTHORITY\NETWORK SERVICE). If not a built-in account, also specify SQL Service Password. This parameter is optional for SQL Server 2008 or 2008 R2.	Win12\Administrator
SQL Service Password	Required: Specify if SQL Service Account is specified. This parameter is optional for SQL Server 2008 or 2008 R2.	●●●
SQL Sysadmin Accounts	Required: A comma-delimited list of user accounts that will be set as system administrators. Each account must either be a local Windows user or a domain user. This parameter is optional for SQL Server 2008 or 2008 R2.	Win12\Administrator
Skip Simulation	If set to "YES", workflow will skip Simulate step and proceed directly to install/upgrade step	NA

2. In the workflow editor, expose any additional parameters that you need. You will specify values for those parameters when you create the deployment or at runtime.
3. Save the changes to the workflow (click **Save** in the lower right corner).
4. Create a new deployment. See "Create a Deployment" in *HPE DMA Quick Start Tutorial* for instructions.
5. On the Parameters tab, specify values (or set the type to Runtime Value) for the required

parameters listed in step 2 and any additional parameters that you have exposed. You do not need to specify values for those parameters whose default values are appropriate for your environment.

6. On the Targets tab, specify one or more targets for this deployment.
7. Save the deployment (click **Save** in the lower right corner).
8. Run the workflow using this deployment, specifying any runtime parameters. See "Run Your Workflow" in (*HPE DMA Quick Start Tutorial* for instructions.

To verify the results:

The workflow will complete and report SUCCESS on the Console if it has run successfully. If an error occurs during workflow execution, the error is logged, and the workflow terminates in the FAILURE state.

Parameters for MS SQL - Add Node to Cluster

The following tables describe the required and optional input parameters for this workflow.

Step: MS SQL - Parameters - Add Node to Cluster

Parameter	Required	Example Value	Description
Cluster Administrator Account	required	Win12\Administrator	The Windows domain user that will run the setup operation. This user requires elevated administrator privileges on the cluster. Format: <DOMAIN>\<USERNAME>
Cluster Administrator Password	required	●●●	Password for the Windows domain user that will run the setup operation. This user requires elevated administrator privileges on the cluster. Must be a strong Windows password.
Download From Software Directory	optional	`SQL12.zip`	The name of the ZIP file that contains the SQL Server installation software files obtained from Microsoft.[1] **Note:** If necessary, manually zip the installation software files up.

[1] If the file is not found on the target server(s), it will be downloaded from the software repository. For additional information, see Alternative methods for specifying input files.

Step: MS SQL - Parameters - Add Node to Cluster, continued

Parameter	Required	Example Value	Description
Download Target Destination	required	`C:\temp`	The local directory where the SQL Server setup files are stored:[1] If source files are in the software repository: Location where Download From Software Directory will be downloaded and extracted If source files are on the target: Location where the Microsoft SQL Server installation files already exist—not zipped up Upon a successful workflow completion, all downloaded and extracted files are cleaned up.
Instance Name	required	SQL-CLUSTER\InstanceA	The name of the newly created virtual server and instance. Format: <Virtual Server>\<Instance Name> Use MSSQLSERVER for the default instance and any other alphanumeric value for a named instance.

Step: MS SQL - Advanced Parameters - Add Node to Cluster

Parameter	Required	Example Value	Description
Additional Template Parameters	optional	SQMREPORTING|1| INSTANCEDIR|"D:\SQLDirectory"	Pipe-delimited (|) list of additional template parameters and values. Should follow this example: SQMREPORTING|1| INSTANCEDIR|"D:\SQLDirectory"
Installer Account	optional	?	The Windows account that will perform the installation.
Installer Password	optional	?	The password of the Windows account that will perform the installation
Product Key	optional	?	Specifies the product key for the

[1]For additional information, see Alternative methods for specifying input files.

Step: MS SQL - Advanced Parameters - Add Node to Cluster, continued

Parameter	Required	Example Value	Description
			edition of SQL Server. If this parameter is not specified, Evaluation is used.
Public IP Address	required	DHCP	Public IP Address. For SQL Server 2012 set to DHCP.
Public IP Network Name	required		IP Network Name for the clusters. Format: <Network Name>:<Subnet Mask> For example: Public:255.255.255.0
SQL Agent Account	optional	Win12\Administrator	The login account for the SQL Server Agent service. Can be a local Windows user, a domain user, or a built-in account (for example, NT AUTHORITY\NETWORK SERVICE). If not a built-in account, also specify SQL Agent Password.
SQL Agent Password	optional	●●●	Specify if SQL Agent Account is specified.
SQL Service Account	optional	Win12\Administrator	The login account for the SQL Server service. Can be a local Windows user, a domain user, or a built-in account (for example, NT AUTHORITY\NETWORK SERVICE). If not a built-in account, also specify SQL Service Password.
SQL Service Password	required	●●●	Required if SQL Service Account is specified and is not a built-in account.
Skip Simulation	optional	NA	If set to "YES", workflow will skip Simulate step and proceed directly to install/upgrade step

MS SQL - Create Database

This workflow creates a new database on the target instance. The only required parameter is “Database Name”, but there are several optional parameters to customize the process.

To use this workflow in your environment, see the following information:

Topic	Information Included
"Prerequisites" below	List of prerequisites that must be satisfied before you can run this workflow
"How this workflow works" on the next page	Information about what the workflow does, including validation checks performed, steps executed, and step descriptions
"How to run this workflow" on page 149	Instructions for running this workflow in your environment
"Parameters for MS SQL - Create Database" on page 153	List of input parameters for this workflow

Prerequisites

Before performing the procedures in this section, your environment must meet the following minimum requirements:

- An existing Windows 2008, 2008 R2, or 2012 cluster
- Installation software:

  The SQL Server 2008, 2008 R2, or 2012 software installation files, obtained from Microsoft.

  The installation media must be available locally or available for download from the software repository.
- Storage:

  An available shared disk for SQL Server shared files

  A staging directory with 4 gigabytes available to unzip the SQL Server software
- Permissions to create an SQL Server database:

  System Stored Procedures (SP)

  CREATE LOGIN

If using a non-default database owner, the `sp_changedbowner` process is available

If a non-default database owner is specified and does not exist, permission to create the appropriate login

- .NET 3.5 is installed.

> **Note:** For additional information, see "Run as a Windows Domain User" in the *HPE DMA Installation Guide*, available at: https://softwaresupport.hp.com/

- Licenses for SQL Server and HPE DMA.

For additional requirements, see the following Microsoft documentation:

SQL Server version	Microsoft documentation
2008	Hardware and Software Requirements for Installing SQL Server 2008
2008 R2	Hardware and Software Requirements for Installing SQL Server 2008 R2
2012	Hardware and Software Requirements for Installing SQL Server 2012

How this workflow works

This workflow creates a new database on the target instance.

Steps Executed

The MS SQL - Create Database workflow includes the following steps. Each step must complete successfully before the next step can start. If a step fails, the workflow reports a failure and all subsequent steps are skipped.

Steps used by MS SQL - Create Database

Workflow Step	Description
MS SQL Parameters Create Database	Gather and validate parameters for Create Database workflow.
MS SQL Advanced Parameters Create Database	Gather and validate optional parameters for Create Database workflow.
MS SQL Check Database Exists	Validates existence of database.
MS SQL Kill Processes	Kills all currently running user processes.

Steps used by MS SQL - Create Database, continued

Workflow Step	Description
MS SQL Validate Directory Paths	Validates a comma-delimited string of directory paths.
MS SQL Drop Database	Drops target database. Ensure that there are no active connections prior to running this step by running the "MS SQL: Kill Processes" step.
MS SQL Validate Directory Paths	Validates a comma-delimited string of directory paths.
MS SQL Verify Server Login	Validates SQL server logins as well as Windows-authenticated server logins.
MS SQL Create Database	Creates a new database on the target Instance.
MS SQL Create Server Login	Validates a comma-delimited string of directory paths.
MS SQL Change Database Owner	Changes owner of target database to specified login.
MS SQL Change Recovery Model	Changes the recovery model of the target database.
MS SQL Set Database Options	This step evaluates a comma-delimited list of option and value pairs, and sets the various database options.
Discover SQL Databases	Audits the server's physical environment looking for SQLServer instances and databases.

How to run this workflow

The following instructions show you how to customize and run the MS SQL - Create Database workflow in your environment.

The workflow provides default values for some parameters. These default values are usually sufficient for a "typical" installation. You can override the defaults by specifying parameter values in the deployment. You can also expose additional parameters in the workflow, if necessary, to accomplish more advanced scenarios. Any parameters not explicitly specified in the deployment will have the default values listed in "Parameters for MS SQL - Create Database" on page 153.

> **Note:** Before following this procedure, review the "Prerequisites" on page 147, and ensure that all requirements are satisfied.

To use the MS SQL - Create Database workflow:

1. Create a deployable copy of the workflow (see "Create a Deployable Workflow" in *HPE DMA Quick Start Tutorial*)
   a. Determine the values that you will specify for the following parameters.

      The following tables describe the required and optional input parameters for this workflow.

Step: MS SQL Parameters Create Database

Parameter	Description	Example Value
Database Name	Required: Name of the new database.	NewDatabase
Web Service Password	Required: Password for the HPE DMA Discovery web service API.	●●●
Web Service User	Required: User who is capable of modifying the managed environment by using the HPE DMA Discovery web service API.	dmawebuser

Step: MS SQL Advanced Parameters Create Database

Parameter	Description	Example Value
Additional Database Options and Values		?
Collation		?
Compatibility Level		?
Data File Paths		?
Data File Sizes,Growths,and Max Sizes	Comma-delimited list of sizing information for each data file. Optional, String. Blank values in list replaced with server defaults (3MB,1MB,0), depending on corresponding value. Values in list are initial size, growth increment, and max size, in that order. First 3 values in list apply to first data file, while next 3 apply to the next data file, and so on. Sizes expressed as [integer] [KB,MB,GB], growth rates	3MB,1MB,0

Step: MS SQL Advanced Parameters Create Database, continued

Parameter	Description	Example Value
	expressed as [integer] [KB,MB,GB,%] or 0 (unlimited).	
Data Filegroups	Comma-delimited list of filegroup (s) associated with data files. Optional, String. Blank values in list replaced with "PRIMARY".	PRIMARY
Database Owner Login Name	Login name of owner of the database. Optional, String. Windows authenticated login format: [domain]\[username]	?
Database Owner Login Password	The password of new owner of database. *Required (if new SQL login needed), String.	?
Database Recovery Model	Database recovery model. Optional, String. Acceptable values = [FULL (default),BULK_ LOGGED,SIMPLE].	?
Default Database		?
Drop Database If Exists	Flag database as droppable if found. Optional,String. Acceptable inputs = YES, NO (default).	NO
Instance Account	Optional: The Windows account that will terminate the SQL Server processes.	
Instance Password	?	?
Log File Path	Directory path, file path, or filename for log file. Optional, String. If blank, replaced with path [server default log directory]\ [dbname]_log.mdf. Acceptable values = directory path (filename [DBname]_log.ldf used), file path, filename (default log directory used), or blank.	?
Log File	Comma-delimited list of sizing	?

Step: MS SQL Advanced Parameters Create Database, continued

Parameter	Description	Example Value
Size,Growth,and Max Size	information for log file. Optional, String. Blank items replaced with server defaults (3MB,1%,0), depending on corresponding value. The values in list are initial size, growth increment, and max size, in that order.	
SQL Instance Account	Optional: Either a Windows or SQL Server user that can log in. Used if the default Windows user is unable to log in to the instance.	?
SQL Instance Password	Optional: The password to the SQL Instance Account. Used if the default Windows user is unable to log in to the instance.	?

2. In the workflow editor, expose any additional parameters that you need. You will specify values for those parameters when you create the deployment or at runtime.
3. Save the changes to the workflow (click **Save** in the lower right corner).
4. Create a new deployment. See "Create a Deployment" in *HPE DMA Quick Start Tutorial* for instructions.
5. On the Parameters tab, specify values (or set the type to Runtime Value) for the required parameters listed in step 2 and any additional parameters that you have exposed. You do not need to specify values for those parameters whose default values are appropriate for your environment.
6. On the Targets tab, specify one or more targets for this deployment.
7. Save the deployment (click **Save** in the lower right corner).
8. Run the workflow using this deployment, specifying any runtime parameters. See "Run Your Workflow" in (*HPE DMA Quick Start Tutorial* for instructions.

To verify the results:

The workflow will complete and report SUCCESS on the Console if it has run successfully. If an error occurs during workflow execution, the error is logged, and the workflow terminates in the FAILURE state.

Parameters for MS SQL - Create Database

The following tables describe the required and optional input parameters for this workflow.

Step: MS SQL Parameters Create Database

Parameter	Required	Example Value	Description
Database Name	required	NewDatabase	Name of the new database.
Web Service Password	required	●●●	Password for the HPE DMA Discovery web service API.
Web Service User	required	dmawebuser	User who is capable of modifying the managed environment by using the HPE DMA Discovery web service API.

Step: MS SQL Advanced Parameters Create Database

Parameter	Required	Example Value	Description
Data File Sizes,Growths,and Max Sizes		3MB,1MB,0	Comma-delimited list of sizing information for each data file. Optional, String. Blank values in list replaced with server defaults (3MB,1MB,0), depending on corresponding value. Values in list are initial size, growth increment, and max size, in that order. First 3 values in list apply to first data file, while next 3 apply to the next data file, and so on. Sizes expressed as [integer][KB,MB,GB], growth rates expressed as [integer][KB,MB,GB,%] or 0 (unlimited).
Data Filegroups		PRIMARY	Comma-delimited list of filegroup(s) associated with data files. Optional, String. Blank values in list replaced with "PRIMARY".
Database Owner Login Name		?	Login name of owner of the database. Optional, String. Windows authenticated login format: [domain]\[username]
Database Owner Login Password		?	The password of new owner of database. *Required (if new SQL login needed), String.
Database Recovery		?	Database recovery model. Optional,

Step: MS SQL Advanced Parameters Create Database, continued

Parameter	Required	Example Value	Description
Model			String. Acceptable values = [FULL (default),BULK_LOGGED,SIMPLE].
Default Database		?	
Drop Database If Exists		NO	Flag database as droppable if found. Optional,String. Acceptable inputs = YES, NO (default).
Instance Account			Optional: The Windows account that will terminate the SQL Server processes.
Instance Password		?	?
Log File Path		?	Directory path, file path, or filename for log file. Optional, String. If blank, replaced with path [server default log directory]\[dbname]_log.mdf. Acceptable values = directory path (filename [DBname]_log.ldf used), file path, filename (default log directory used), or blank.
Log File Size,Growth,and Max Size		?	Comma-delimited list of sizing information for log file. Optional, String. Blank items replaced with server defaults (3MB,1%,0), depending on corresponding value. The values in list are initial size, growth increment, and max size, in that order.
SQL Instance Account		?	Optional: Either a Windows or SQL Server user that can log in. Used if the default Windows user is unable to log in to the instance.
SQL Instance Password		?	Optional: The password to the SQL Instance Account. Used if the default Windows user is unable to log in to the instance.

Send documentation feedback

If you have comments about this document, you can contact the documentation team by email. If an email client is configured on this system, click the link above and an email window opens with the following information in the subject line:

Feedback on Workflows for SQL Server (Database and Middleware Automation 10.50)

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to hpe_dma_docs@hpe.com.

We appreciate your feedback!