

Microsoft System Center 2012 R2

Administering System Center 2012 R2 Orchestrator

Microsoft Corporation

Published: November 1, 2013

Applies To

System Center 2012 SP1

System Center 2012 R2 Orchestrator

Feedback

Send suggestions and comments about this document to sc2012docs@microsoft.com.

Copyright Information

This document is provided "as-is". Information and views expressed in this document, including URL and other Internet website references, may change without notice.

Some examples depicted herein are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes. You may modify this document for your internal, reference purposes.

© 2013 Microsoft Corporation. All rights reserved.

Microsoft, Access, Active Directory, ActiveSync, ActiveX, Authenticode, Bing, BitLocker, Excel, Forefront, Hyper-V, Internet Explorer, JScript, Microsoft Press, MSDN, Outlook, SharePoint, Silverlight, SoftGrid, SQL Server, Visio, Visual Basic, Visual C++, Visual Studio, Win32, Windows, Windows Intune, Windows Mobile, Windows PowerShell, Windows Server, Windows Server System, and Windows Vista are trademarks of the Microsoft group of companies. All other trademarks are property of their respective owners.

Revision History

Release Date	Changes
October 17, 2013	Original release of this guide.
November 1, 2013	Minor updates for this guide.

Contents

Administering System Center 2012 - Orchestrator.....	4
Accessibility for People with Disabilities	5
Accessibility Features of Orchestrator	5
Accessibility Features of Orchestrator Help	6
Accessibility Products and Services from Microsoft	9
How to Change the Orchestrator Database	10
How to Change the Orchestrator Users Group	12
How to Configure Orchestrator Database Connections.....	14
How to Configure Runbook Servers to Optimize Performance of .NET Activities	14
How to Configure Runbook Throttling	15
How to View Orchestrator Data by Using PowerPivot	16
Orchestrator Logs.....	19
Runbook logs	20
Activity Events.....	24
Audit Trail.....	26
Trace Logs	30
Runbook Server Properties.....	31
How to Back up Orchestrator	32
Migrate Orchestrator Between Environments	33
Best Practices Analyzer	35
How to Benchmark.....	39
How to Recover a Database	42
Database Sizing and Performance	44
How to Recover Web Components.....	52
Feature Performance Considerations	53
How to Use the Integration Toolkit to Extend Orchestrator Capabilities.....	54

Administering System Center 2012 - Orchestrator

System Center 2012 - Orchestrator is a workflow management solution for the datacenter that you can use to automate the creation, monitoring, and deployment of resources in your environment. This guide describes tasks that an administrator might have to perform to maintain a production Orchestrator environment. It assumes that you have a fully deployed Orchestrator environment as described in **Deploying System Center 2012 - Orchestrator Release Candidate**.

In This Guide

- [Accessibility for People with Disabilities](#)
Describes accessibility options in Orchestrator.
- [How to Change the Orchestrator Database](#)
Describes how to change the location of the orchestration database.
- [How to Change the Orchestrator Users Group](#)
Describes how to remove and add members to the Orchestrator Users Group.
- [How to Configure Orchestrator Database Connections](#)
Describes how to use DBSetup to configure connections to an Orchestrator database.
- [How to Configure Runbook Servers to Optimize Performance of .NET Activities](#)
Describes how to configure a runbook server to perform optimally with activities that require Microsoft .NET libraries.
- [How to Configure Runbook Throttling](#)
Describes how to use the Runbook Server Runbook Throttling tool to change the maximum number of runbooks that can run on a runbook server.
- [How to View Orchestrator Data by Using PowerPivot](#)
Describes how to view Orchestrator data by using Microsoft Excel PowerPivot.
- [Orchestrator Logs](#)
Describes the types of logs available in Orchestrator and how to configure them.
- [Runbook Server Properties](#)
Describes the runbook server properties.
- [How to Back up Orchestrator](#)
Describes how to back up an Orchestrator database.
- [How to Use the Integration Toolkit to Extend Orchestrator Capabilities](#)
Provides a link to the MSDN library content that helps you create new integration packs for Orchestrator.

Other Resources for this Product

- TechNet Library main page for **Orchestrator Release Candidate**
- **Getting Started with System Center 2012 - Orchestrator Release Candidate**
- **Deploying System Center 2012 - Orchestrator Release Candidate**
- **Integration Packs for System Center 2012 - Orchestrator Release Candidate**
- **Using Runbooks in System Center 2012 - Orchestrator Release Candidate**
- **Using the Orchestration Console in System Center 2012 - Orchestrator Release Candidate**
- **Runbook Activity Reference for System Center 2012 - Orchestrator Release Candidate**

Accessibility for People with Disabilities

Microsoft is committed to making its products and services easier for everyone to use. The following topics provide information about the features, products, and services that make Microsoft System Center 2012 - Orchestrator more accessible for people with disabilities.

In This Section

- [Accessibility Features of Orchestrator](#)
Describes the accessibility features of Orchestrator.
- [Accessibility Features of Orchestrator Help](#)
Describes the accessibility features of the Orchestrator Help.
- [Accessibility Products and Services from Microsoft](#)
Describes the accessibility products and services that are available from Microsoft.

See Also

Orchestrator Release Candidate

Accessibility Features of Orchestrator

In addition to accessibility features and tools in Windows, the following feature makes Orchestrator more accessible for people with disabilities.

For information about using the Orchestrator Command Line Install tool, see **Install with the Orchestrator Command Line Install Tool** in **Deploying System Center 2012 - Orchestrator Release Candidate**.



Note

The information in this section may apply only to users who license Microsoft products in the United States. If you obtained this product outside of the United States, you can use the subsidiary information card that came with your software package or visit the [Microsoft Accessibility website](#) for a list of Microsoft support services telephone numbers

and addresses. You can contact your subsidiary to find out whether the type of products and services described in this section are available in your area. Information about accessibility is available in other languages, including Japanese and French.

Using Keyboard Shortcuts in the Orchestration Console

To do this	Use this keyboard shortcut
Navigate to a control.	Tab or any arrow key
Activate the edit mode.	Enter
Exit the edit mode.	Esc
Select a control.	Enter
Browse runbooks.	Ctrl+Alt+B
Browse runbook servers.	Ctrl+Alt+V
Browse events.	Ctrl+Alt+E
View summary.	Ctrl+Alt+S
View runbooks.	Ctrl+Alt+R
View jobs.	Ctrl+Alt+J
View instances.	Ctrl+Alt+I
View privacy policy.	Ctrl+Alt+P
Provide feedback.	Ctrl+Alt+F
Open help.	Ctrl+Alt+H
Open query menu.	Ctrl+Alt+Q
Start runbook.	Ctrl+Alt+Y
Stop runbook.	Ctrl+Alt+Z

See Also

[Accessibility for People with Disabilities](#)

Accessibility Features of Orchestrator Help

Orchestrator Help includes features that make it accessible to a wider range of users, including those who have limited dexterity, low vision, or other disabilities. In addition, Orchestrator Help is available on the Web at [Accessibility for People with Disabilities](#).

Keyboard Shortcuts for Using the Help Window

By using the following keyboard shortcuts in Help, you can quickly accomplish many common tasks.

To do this	Use this keyboard shortcut
Display the Help window.	F1
Switch the cursor between the Help topic pane and the navigation pane (tabs such as Contents , Search , and Index).	F6
Change between tabs (for example, Contents , Search , and Index) while in the navigation pane.	ALT + Underlined letter of the tab
Select the next hidden text or hyperlink.	TAB
Select the previous hidden text or hyperlink.	SHIFT+TAB
Perform the action for the selected Show All, Hide All, hidden text, or hyperlink.	ENTER
Display the Options menu to access any Help toolbar command.	ALT+O
Hide or show the pane containing the Contents , Search , and Index tabs.	ALT+O, and then press T
Display the previously viewed topic.	ALT+O, and then press B
Display the next topic in a previously displayed sequence of topics.	ALT+O, and then press F
Return to the specified home page.	ALT+O, and then press H
Stop the Help window from opening a Help topic (useful if you want to stop a Web page from downloading).	ALT+O, and then press S
Open the Internet Options dialog box for Windows Internet Explorer, where you can change accessibility settings.	ALT+O, and then press I
Refresh the topic (useful if you have linked to a Web page).	ALT+O, and then press R
Print all topics in a book or a selected topic only.	ALT+O, and then press P
Close the Help window.	ALT+F4

Procedures

▶ To change the appearance of a Help topic

1. To customize the colors, font styles, and font sizes used in Help, open the Help window.
2. Click **Options**, and then click **Internet Options**.
3. On the **General** tab, click **Accessibility**. Select **Ignore colors specified on Web pages**, **Ignore font styles specified on Web pages**, and **Ignore font sizes specified on Web pages**. You also can choose to use the settings specified in your own style sheet.
4. To change the colors used in Help, see "To change the color of the background or text in Help". To change the font, see "To change the font in Help."

▶ To change the color of the background or text in Help

1. Open the Help window.
2. Click **Options**, and then click **Internet Options**.
3. On the **General** tab, click **Accessibility**. Then, select **Ignore colors specified on Web pages**. You also can choose to use the settings specified in your own style sheet.
4. To customize the colors used in Help, on the **General** tab, click **Colors**. Clear the **Use Windows Colors** check box, and then select the font and background colors that you want to use.



Note

If you change the background color of the Help topics in the Help window, the change also affects the background color when you view a Web page in Windows Internet Explorer.

▶ To change the font in Help

1. Open the Help window.
2. Click **Options**, and then click **Internet Options**.
3. On the **General** tab, click **Accessibility**. To use the same settings as those used in your instance of Windows Internet Explorer, select **Ignore font styles specified on Web pages** and **Ignore font sizes specified on Web pages**. You also can choose to use the settings specified in your own style sheet.
4. To customize the font style used in Help, on the **General** tab, click **Fonts**, and then click the font style you want.



Note

If you change the font of the Help topics in the Help window, the change also affects the font when you view a Web page in Internet Explorer.

Accessibility Products and Services from Microsoft

Microsoft is committed to making its products and services easier for everyone to use. The following sections provide information about the features, products, and services that make Microsoft® Windows® more accessible for people with disabilities:

- Accessibility Features of Windows
- Documentation in Alternative Formats
- Customer Service for People with Hearing Impairments
- For More Information



Note

The information in this section may apply only to users who license Microsoft products in the United States. If you obtained this product outside of the United States, you can use the subsidiary information card that came with your software package or visit the [Microsoft Accessibility website](#) for a list of Microsoft support services telephone numbers and addresses. You can contact your subsidiary to find out whether the type of products and services described in this section are available in your area. Information about accessibility is available in other languages, including Japanese and French.

Accessibility Features of Windows

The Windows operating system has many built-in accessibility features that are useful for individuals who have difficulty typing or using a mouse, are blind or have low vision, or who are deaf or hard-of-hearing. The features are installed during Setup. For more information about these features, see Help in Windows and the [Microsoft Accessibility website](#).

Free Step-by-Step Tutorials

Microsoft offers a series of step-by-step tutorials that provide detailed procedures for adjusting the accessibility options and settings on your computer. This information is presented in a side-by-side format so that you can learn how to use the mouse, the keyboard, or a combination of both. To find step-by-step tutorials for Microsoft products, see the [Microsoft Accessibility website](#).

Assistive Technology Products for Windows

A wide variety of assistive technology products are available to make computers easier to use for people with disabilities. You can search a catalog of assistive technology products that run on Windows at the [Microsoft Accessibility website](#). If you use assistive technology, be sure to contact your assistive technology vendor before you upgrade your software or hardware to check for possible compatibility issues.

Documentation in Alternative Formats

If you have difficulty reading or handling printed materials, you can obtain the documentation for many Microsoft products in more accessible formats. You can view an index of accessible product documentation on the [Microsoft Accessibility website](#). In addition, you can obtain additional Microsoft publications from Learning Ally (formerly Recording for the Blind & Dyslexic,

Inc.). Learning Ally distributes these documents to registered, eligible members of their distribution service. For information about the availability of Microsoft product documentation and books from Microsoft Press, contact:

Learning Ally (formerly Recording for the Blind & Dyslexic, Inc.)

20 Roszel Road
Princeton, NJ 08540

Telephone number from within the United States: (800) 221-4792

Telephone number from outside the United States and Canada: (609) 452-0606

Fax: (609) 987-8116

[Learning Ally website](#)

Web addresses can change, so you might be unable to connect to the website or sites mentioned here.

Customer Service for People with Hearing Impairments

If you are deaf or hard-of-hearing, complete access to Microsoft product and customer services is available through a text telephone (TTY/TDD) service:

- For customer service, contact Microsoft Sales Information Center at (800) 892-5234 between 6:30 AM and 5:30 PM Pacific Time, Monday through Friday, excluding holidays.
- For technical assistance in the United States, contact Microsoft Product Support Services at (800) 892-5234 between 6:00 AM and 6:00 PM Pacific Time, Monday through Friday, excluding holidays. In Canada, dial (905) 568-9641 between 8:00 AM and 8:00 PM Eastern Time, Monday through Friday, excluding holidays.

Microsoft Support Services are subject to the prices, terms, and conditions in place at the time the service is used.

For More Information

For more information about how accessible technology for computers helps to improve the lives of people with disabilities, see the [Microsoft Accessibility website](#).

How to Change the Orchestrator Database

You might have to change the location of the Orchestrator database after installation, because you might want to separate the management server and database server, move the database to a larger server or a cluster, or just reconfigure the orchestration database based on required changes in your environment. You can use standard Microsoft SQL Server methods to move the existing database to another server, but then you must configure the Orchestrator features to connect to the new server. You must perform this configuration for the management server, the

web service supporting the Orchestration console, and each runbook server as described in the following procedures.

Management server and runbook servers

You can use the Database Configuration utility to change the connection settings that the management server and runbook servers installed in your environment. The settings for these servers are stored in an encrypted file called **Settings.dat**. If you change your orchestration database settings, such as the port, user account access, or computer name, you must manually uninstall and reinstall all runbook servers, and then re-run the Database Configuration utility on the management server and all runbook servers.

► To change the database settings for the management server and runbook servers

1. On the management server, click **Start**, point to **All Programs**, click **Microsoft System Center 2012**, click **Orchestrator**, and then click **Data Store Configuration**.
2. In the **Server** box, enter the name of the server that is hosting the database by using the format **<server>\<instance>,<port>**. You can click the ellipsis (...) button to select the computer. You do not have to include the instance if the Orchestrator database is installed on the default instance. You do not have to include the port if SQL Server is usually installed on the default port 1433.

If the Orchestrator database is installed on an instance called MyInstance on a computer named MySQLServer that is configured on port 12345, enter **MySQLServer\MyInstance,12345**.

If the Orchestrator database is installed on an instance called MyInstance on a computer named MySQLServer that is configured on port 1433, enter **MySQLServer\MyInstance**.

If the orchestration database is installed on the default instance on a computer named MySQLServer that is configured on port 1433, enter **MySQLServer**.

3. Select the authentication method to use to connect to the SQL Server:
 - **Windows Authentication** Connect to the SQL Server by using Windows Authentication.
 - **SQL Server Authentication** Connect to the SQL Server by using a SQL Server user account. Type the **User Name** and **Password** of the SQL Server user account. This account must have rights to create, write, and own a database and create, update, and delete rows in the database.
4. Click **Next**.
5. In the **Data Store** pane, click **Use an existing database**.
6. In the **Name** list, select the database.
7. Click **Finish**.

Web Service

The web service supporting the Orchestration console does not use the **Settings.dat** file. To change the database settings for the web service, you must modify the Web.config file on the Internet Information Services (IIS) server. You can use **IIS Manager** to modify the file, but you must first decrypt it by running the aspnet_regiis.exe executable file.

▶ To change the database settings for the Orchestrator web service

1. Log on with administrative credentials to the computer with the Orchestration console installed.
2. Open a Command Prompt window with administrator credentials.
3. Run the following command to decrypt the Web.config file:
C:\Windows\Microsoft.NET\Framework\v4.0.30319\aspnet_regiis.exe -pdf "connectionStrings" "C:\Program Files (x86)\Microsoft System Center 2012\Orchestrator\Web Service\Orchestrator2012"
4. To start the IIS Manager, click **Start**, point to **Administrative Tools**, and then click **Internet Information Services (IIS) Manager**.
5. Expand the **Sites** node, and then click **Microsoft System Center 2012 Orchestrator Web Service**.
6. In the **Features View**, double-click **Connection Strings**.
7. In the **Connections String** pane, double-click **OrchestratorContext**.
8. In the **Custom** box, scroll down to the portion of the string that includes the server name (Data Source) and database name (Initial Catalog). Modify these values as required.
9. Click **OK** to close the dialog box.
10. Close **IIS Manager**.
11. Run the following command to encrypt the Web.config file:
C:\Windows\Microsoft.NET\Framework\v4.0.30319\aspnet_regiis.exe -pef "connectionStrings" "C:\Program Files (x86)\Microsoft System Center 2012\Orchestrator\Web Service\Orchestrator2012"

See Also

[Administering System Center 2012 - Orchestrator](#)

How to Change the Orchestrator Users Group

You might want to change the Orchestrator users group after installation because of changes in your environment. For example, you might want to use a local group during installation, and then change it to a domain account later.

PermissionsConfig tool

You can change the Orchestrator Users group by using the PermissionsConfig tool, which is located on the management server in **<InstallDir>Management Server**. The syntax of this tool is as follows:

PermissionsConfig–OrchestratorUsersGroup *GroupName*–**OrchestratorUser** *UserName*–**remote**

Note that the PermissionsConfig tool does not send results to standard output. To view the results of the command, check the **%errorlevel%** in the Orchestrator log file that is located at **C:\Users\SCXSVC\AppData\Local\SCO\LOGS**. The results are 1 for failure, 0 for success.

You can get an explanation of the parameters for the PermissionsConfig tool by typing the following command:

```
PermissionsConfig -help
```

The following table explains the parameters.

Parameter	Details
OrchestratorUsersGroup	The name of the group to use for Orchestrator permissions.
OrchestratorUser	If this parameter is specified with a user name, the user is granted immediate access to Orchestrator whether a member of the specified group or not. This is to prevent the requirement for the user to log off and on if the group has just been created.
Remote	Indicates that the Runbook Designer can be run from a computer other than the management server.

For example, to change the Orchestrator users group to a group that is named Orchestrator Users in a domain that is named Contoso, use the following command:

```
PermissionsConfig -OrchestratorUsersGroup "Contoso\Orchestrator Users" -remote
```

Important

You must run the PermissionsConfig tool at a command prompt with administrative credentials because it modifies group memberships. To do this, right-click the **Command Prompt** icon to select **Run as Administrator**.

See Also

Orchestrator Security Planning

How to Configure Orchestrator Database Connections

DBSetup allows you to secure an unsecured database. The common scenario is connecting to a restored backup.

This utility provides two functions:

1. DBSetup allows you to change the database name or credentials that are used by the management server or runbook servers to connect to the database.
2. DBSetup allows you to connect to a rebuilt database.

When connecting to a rebuilt database:

- This procedure can only be performed against the same database server used during the installation of the management server.
- You must have database permissions to create the database.

In contrast, DBconfig only creates a new database; it does not configure the security for the database. DBConfig configures the database schema in the database and creates the contents of **settings.dat**, which contains the connection details for the management server and runbook servers. For more information on running DBConfig, see [How to Change the Orchestrator Database](#).

▶ To configure Orchestrator database connections

- Run the DBsetup binary from the **Start** menu or from the **Program Files** folder.

▶ To create a new database on a new database server

1. Run the System Center 2012 - Orchestrator Setup Wizard and install a new management server.
2. On the **Configure the database server** page in the setup wizard, point to the new database server.
3. After you add a new DB server to your deployment, you must also run permissionsconfig, and then export and import the service master key to the new database server.

See Also

[How to Change the Orchestrator Database](#)

How to Configure Runbook Servers to Optimize Performance of .NET Activities

If a runbook contains an activity that references the .NET libraries, the first reference to the .NET libraries takes additional time to initialize. This delay can be as much as 30 seconds. All remaining activities that reference the .NET libraries run immediately. This delay can also occur when a runbook is started on a computer without Internet access, because then Windows cannot

verify the Microsoft Authenticode signature for the .NET libraries, and this causes a delay during the initialization of the activity.

The solution to removing the delay is to deactivate **generatePublisherEvidence** in **PolicyModule.exe** or to create a profile for the service account.

▶ **To deactivate generatePublisherEvidence in policymodule.exe.config**

1. On the runbook server where runbooks that contain an activity referencing the .NET libraries run, locate the file **C:\Program Files (x86)\Microsoft System Center 2012\Orchestrator\Runbook Server\policymodule.exe.config**.
2. Add the following code to policymodule.exe.config:

```
<runtime>
    <generatePublisherEvidence enabled="false"/>
</runtime>
```

▶ **To create a profile for the service account**

- On the runbook server where runbooks run that contain an activity referencing the .NET libraries, log on to the computer that is using the service account credentials. A profile is created on first logon.

See Also

[Administering System Center 2012 - Orchestrator](#)

How to Configure Runbook Throttling

By default, each runbook server is configured to simultaneously run a maximum of 50 runbooks. You can change this number by using the Runbook Server Runbook Throttling tool. In most cases, you can keep this default setting, but you should consider the resource requirements of the runbooks on a particular server when considering whether to change it. If the server has a number of runbooks with high resource requirements, you might run fewer runbooks simultaneously on the runbook server. If they are simple runbooks with minimal requirements, you might consider increasing the number of simultaneously run runbooks.

▶ **To configure the maximum number of runbooks that a runbook server processes**

1. Navigate to the folder where by default the Runbook Server Runbook Throttling tool is stored: **<System Drive>:\Program Files (x86)\Microsoft System Center 2012\Orchestrator\Management Server**.
2. Type one of the following commands:
 - To apply the change to one runbook server:
aspt <RunbookServerName> <MaximumRunningRunbooks>.
For example, to set the maximum number of runbooks that RunbookServer1 runs to

40:

aspt RunbookServer1 40

- To apply the change to all runbook servers:

aspt * <MaximumRunningRunbooks>.

For example, to set the maximum number of runbooks that all runbook servers run to 40:

aspt * 40

3. Restart the **Orchestrator Runbook Service**.

See Also

[Administering System Center 2012 - Orchestrator](#)

How to View Orchestrator Data by Using PowerPivot

You can use Microsoft PowerPivot for Microsoft Excel to create reports for System Center 2012 - Orchestrator. You configure PowerPivot to use the Orchestrator web service as a data feed, filter the Source Tables for the data you want to use, and import the tables into the PowerPivot worksheet. PowerPivot lets you create relationships between tables, and manipulate the data to fit your requirements. By using the PivotTable feature in PowerPivot, you can generate a report that uses any of the data contained within the PowerPivot workbook.

Connect the Orchestrator web service to PowerPivot for Excel

You must install PowerPivot for Excel to enable the product.

PowerPivot for Excel requires Excel 2010 (64-bit or 32-bit).

To install PowerPivot

1. Follow the instructions found at [Install PowerPivot for Excel](#).

Use PowerPivot to configure a connection to Orchestrator web service. Orchestrator uses the Open Data Protocol (OData), which PowerPivot can consume.



Note

The OData provider in PowerPivot does not support the data contained in the Runbook Diagram box. Attempts to add a Runbook Diagram table will fail.

To create a connection to an Orchestrator feed

1. Open Excel.
2. Click the **PowerPivot** tab above the ribbon.

3. Click **PowerPivot Window** on the ribbon. A **PowerPivot for Excel** book opens.
4. Click **From Data Feeds** on the ribbon. A **Table Import Wizard** opens.
5. Enter the Orchestrator web service URL in the **Data Feed URL** box. The web service URL is on port 81 of the Orchestrator SQL Server. For example, `http://orchestrator:81/Orchestrator2012/Orchestrator.svc`.
6. Click **Test Connection**.
7. If the test connection is successful, click **OK** and proceed to the next step.
If the test connection fails, do the following:
 - a. Click **OK**.
 - b. Click **Advanced**. The **Advanced** dialog box opens.
 - c. In the **Security** section, change **Integrated Security** to **Basic**.
 - d. Change **Persist Security Info** to **True**.
 - e. Enter your **User ID** and **Password** in the appropriate boxes.
 - f. Click **Test Connection**.
 - g. Click **OK** and click **OK**.
8. Click **Next**.
9. Select the check boxes of the table or tables that you want to import.
10. To filter columns, select a table, click **Preview & Filter**, clear any boxes to exclude, and then click **OK**.
11. Click **Finish**. The data is imported.
12. Click **Close**.

Create a Summary of Runbook Results

The following procedure describes the steps to create a pivot table containing a list of all runbooks and the count of results, grouped by the runbook server that ran the runbook instance.



Note

For this example, the orchestration database must contain results from at least one runbook for PowerPivot to import a table.

▶ To create a connection to the data feed

1. Open Excel.
2. Click the **PowerPivot** tab above the ribbon.
3. Click **PowerPivot Window** on the ribbon. A **PowerPivot for Excel** book opens.
4. Click **From Data Feeds** on the ribbon. A **Table Import** wizard opens.
5. Enter the Orchestrator web service URL in the **Data Feed URL** box.
6. Click **Next**.
7. Select the check boxes of the **Runbooks**, **RunbookInstances**, and **RunbookServers** tables.

8. Click **Finish**. The data is imported.
9. Click **Close**.

▶ **To create relationships in PowerPivot**

1. In the **PowerPivot for Excel** window, select the **RunbookInstance** tab.
2. Right-click the header of the **RunbookId** column to select **Create Relationship**.
3. In the **Related Lookup Table** list, select **Runbooks**, and in the **Related Lookup Column** list, select **Id**, and then click **Create**.
4. Right-click the header of the **RunbookServerId** column to select **Create Relationship**.
5. In the **Related Lookup Table** list, select **RunbookServers**, and in the **Related Lookup Column** list, select **Id**, and then click **Create**.

For additional information about PowerPivot relationships, see [Introduction to PowerPivot relationships](#).

▶ **To create a pivot table**

1. In the **PowerPivot for Excel** window, click **PivotTable** on the ribbon, and select **PivotTable**.
2. In the **Create PivotTable** dialog box, select **New Worksheet**, and then click **OK**.
3. In the **PowerPivot Field List**, under **RunbookServers**, click and drag **Name** to the **Row Labels** box.
4. In the **PowerPivot Field List**, under **Runbooks**, click and drag **Name** to the **Row Labels** box.
5. In the **PowerPivot Field List**, under **RunbookInstances**, click and drag **Status** to the **Column Labels** box.
6. In the **PowerPivot Field List**, under **RunbookInstances**, click and drag **RunbookId** to the **Sum Values** box.
7. Right-click **RunbookId** to select **Summarize by**, and then click **Count**.

You can now modify the default labels and format your table for presentation.

For more information about the workflow of a runbook and an explanation of runbook jobs and runbook instances, see **Orchestrator Architecture** in the **Getting Started with System Center 2012 - Orchestrator Release Candidate**.

For more information about PowerPivot for Excel, see [Introducing PowerPivot for Excel](#).

See Also

[Administering System Center 2012 - Orchestrator](#)

Orchestrator Logs

This section provides information on common troubleshooting issues and the available tools that can assist in identifying root problems.

Log files

In Orchestrator, different logs are available that provide information about Orchestrator runbooks and servers. The following table lists the available types of log files, with links to the appropriate sections that describe the logs in more detail.

Log File Topic or Section	Contents	Where data is stored	Where data is viewed
Real time log section in Runbook Logs.	Live information about a running runbook instance	Orchestration database	Log tab in Runbook Designer
Historic Log section in Runbook Logs.	Historical information about instances of a runbook	Orchestration database	Log History tab in Runbook Designer
Runbook Audit History section in Runbook Logs.	Audit information about changes to a runbook	Orchestration database	Audit History tab in Runbook Designer
Activity Events	Status information about Orchestrator management server, runbook servers, and database	Orchestration database	Events tab in Runbook Designer
Audit Trail	Interaction of a runbook with external tools and systems	Log files	Open files in text editor
Trace Logs	Troubleshooting information about the Orchestrator environment	Log files	Open files in text editor

Other resources for this product

- TechNet Library main page for **System Center Orchestrator 2012**
- **System Center Orchestrator 2012 Runbook Guide**
- **Designing a Runbook**
- **Building a Runbook**

- **How to Test a Runbook**

Runbook logs

Every runbook generates a log when it is running. The Runbook Designer enables you to view both real-time log information and historic logs.

Real-Time Log

You can view the real-time log of a runbook on the **Log** tab in the **Log** pane of the Runbook Designer. The log has an entry for each running instance of the runbook and the time that it was started. If you expand this entry, you can view the activity that is currently running. Double-click the activity to view its details. If you have configured logging for the workbook to include Published Data, this information is included in the activity's details.

Historic Log

You can view the historic log of a runbook on the **Log History** tab in the **Log** pane of the Runbook Designer. This log has an entry for each completed instance of the runbook with its start and end times and completion status. If you double-click an entry, the **Runbook Details** dialog box opens that includes each activity in the runbook and its completion status. Double-click each activity to view its details. If you have configured logging for the workbook to include Published Data, this information is included in the activity's details.

Runbook Log Options

By default, the Runbook logs do not include Published Data for each activity. For standard activities, refer to **Standard Activities** for a list of the Published Data items available for each. For integration packs, refer to the specific integration pack guide. You can change the logging properties for an individual runbook to include this information in the logs. You can include both or either of the following two kinds of Published Data:

- *Activity-specific Published Data* is Published Data that is specific to a particular activity. Typically, this option should be enabled only for debugging purposes.
- *Common Published Data* is a set of data items that are common to all activities. These are as follows:
 - Activity Name
 - Activity Type
 - Activity ID
 - Activity End Time Year, Month, Day, Weekday, Hours, Minutes, Seconds
 - Activity Duration
 - Previous Activity
 - Previous Activity Name
 - Time Published Data

 **Caution**

By default, logging options are disabled. Be aware that adding these items to the log increases the size of the data store.

Runbook Audit History

The **Runbook Audit History** tracks the changes to each activity in a runbook. This includes the user that made the change, the date and time that the change occurred, the activity that was changed, and the current and previous values of any changed properties. The audit history is maintained as long as the runbook exists and cannot be cleared.

You can view the **Runbook Audit History** on the **Audit History** tab of the Runbook Designer. The entries for the currently selected runbook will be displayed. Double-click an entry to open the **Details** dialog box where you can view a list of all activities that were part of the change. Select one of the entries to view the activity that was performed in addition to the old and new values of any properties that were changed.

Purging Runbook Logs

The **Runbook Designer** provides a **Log Purge** feature. Orchestrator log data can be purged on demand, or on a scheduled basis. A good practice is to schedule a log purge regularly during off-peak hours to maintain the database.

 **Note**

When logs are purged, the data is deleted from Orchestration database and cannot be recovered.

If you let the number of log entries grow indiscriminately, it can result in excessive storage requirements for the Orchestration database. To maintain the health and performance of the database, you should regularly purge old log entries. You can purge either them on an automatic schedule or manually.

Automatically Purging Runbook Logs

You can schedule the runbook logs to be purged on a regular basis by selecting frequency options and how many entries to retain. By default, runbook logs are automatically purged with the following settings.

Setting	Default value
Frequency	One time per day
Time	2:00 AM
Retain	Keep last 5000 entries

The default log purge job purges all but the last 500 log entries per runbook. This means that if an Orchestrator deployment had 20 runbooks, the default purge would keep the last 500 log entries for each runbook, for a maximum of 10,000 log entries for the database for all runbooks.

You can continue to use these defaults or change the settings by using the following process in the Runbook Designer.

▶ **To purge runbooks on demand or set the purge frequency**

1. In the **Runbook Designer**, right-click **Management server** to select **Log Purge**.
If you want to purge all the runbooks now, click **Purge Now**.
If you want to purge a specific runbook now, click the **Log History** tab to select the runbook to purge.
2. Ensure that the **Schedule Log Purge** option is selected.
3. Select the number of days between running the purge process and a time to run the process.
4. If you do not want to delete all entries, set the **Log Purge Options** to specify the log entries that you want to keep.
5. Click **Finish**.

Manually Purging the Runbook Logs

Audit History logs are a special case because these logs cannot be purged with the **Log Purge** feature. The audit history feature tracks the changes made to a runbook in the **Runbook Designer** and cannot be deleted. The only way to delete Audit History logs is to delete the runbook associated with them. However, data volumes for the Audit History logs for a given runbook are generally small, even in large deployments storage will not require more than the minimum system requirements.

There are two options to manually purge the runbook logs as shown in the following two procedures. They are performed in the Runbook Designer.

▶ **To manually purge the runbook logs for all runbooks**

1. Right-click **Management server** to select **Log Purge**.
2. Set the **Log Purge Options** to specify the log entries that you want to keep.
3. Click **Purge Now**.

▶ **To manually purge all runbook log entries for a single runbook**

1. In the **Connections** pane, select the **Runbooks** folder.
2. In the workspace pane, click the tab of the runbook.
3. In the **Log History** pane, click the **Log History** tab.
4. Click the **Recycle Bin** icon at the top of the pane.
5. When prompted whether you are sure that you want to purge the logs, click **Yes**.

▶ **To delete a specific set of runbook log entries for a single runbook**

1. In the **Connections** pane, select the **Runbooks** folder.
2. In the workspace pane, click the tab of the runbook.
3. In the **Log History** pane, click the **Log History** tab.
4. Select a single log entry or use the key combination Ctrl+Shift to select multiple entries.
5. Click the red **Delete** icon at the top of the pane.
6. When prompted if you are sure that you want to delete the selected entries, click **Yes**.

Removing Orphaned Log Entries

Orphaned log files can be left in the orchestration database if the Runbook Server Service is stopped suddenly while runbook instances are running. If that happens, the runbook instance status does not get updated correctly with its ending time. These entries are removed when the log entries are purged by using one of the purging methods described above. If you want to remove orphaned entries before the next time that the entries are purged, you can manually run the **ClearOrphanedRunbookInstances** stored procedure to perform this function.

The following process describes how to run this stored procedure by using the Microsoft SQL Server Management Studio installed on the orchestration database server. You can perform this process from any computer with this tool installed that has access to the orchestration database.

▶ **To remove orphaned log entries**

1. On the server with the orchestration database, click **Start**, point to **All Programs**, click **Microsoft SQL Server 2008**, and then click **SQL Server Management Studio**.
2. In the **Connect to Server** dialog box, in the **Server name** list, select **(local)**, and then click **Connect**.
3. In the **Object Explorer**, expand **Databases**, and then click **Orchestrator**.

 **Note**

If you used a different name for the Orchestrator database, select that name.

4. Expand **Programmability**, and then click **Stored Procedures**.
5. Right-click **Microsoft.SystemCenter.Orchestrator.Runtime.Internal.ClearOrphanedRunbookInstances** to select **Execute Stored Procedure**.
6. In the **Execute Procedure** dialog box, click **OK**.
7. On the toolbar, click **Execute**.
8. When you receive a message at the bottom of the Logging pane that the query has completed, close **SQL Server Management Studio**.

See Also

[Orchestrator Logs](#)

Activity Events

Each activity in an Orchestrator runbook has the ability to send an event whenever it fails to run or is taking too long to run. These events are presented on the **Events** tab of the Runbook Designer or can be configured to be delivered to a receiver as an SNMP trap. Runbook activity events are only sent for those activities that you specifically configure to do so.

▶ To configure an activity to send events

1. Open the runbook in the **Runbook Designer**.
2. Double-click the activity to view its properties.
3. Click the **Run Behavior** tab.
4. Type a number of seconds to send an event when the activity runs too long.
5. Check the **Report if the activity fails to run** box to send an event when the activity fails.
6. Click **Finish** to save the activity.

Receiving Events from SNMP

In addition to viewing the events on the **Events** tab in the Runbook Designer, you can send them to an SNMP trap destination. This lets you monitor the health of the Orchestrator environment by using other tools designed to provide proactive alerting. The only requirement for such a tool is that it can receive SNMP traps. You can use the **Orchestrator Event Delivery Configuration Utility** to add and configure SNMP trap destinations for Runbook events.

Add an SNMP Trap Destination

To add an SNMP trap destination, run the **oedc** command one time for each destination that you want to add by using the following syntax:

```
oedc /snmp /add /ip <Targeted IP Address> /port <Targeted Port> /version <version> /community <community>
```

For example, use the following procedure to send traps by using SNMP version 1 to an SNMP receiver at IP address 10.1.1.10 on port 162 and a community called public.

▶ To add an SNMP trap destination

1. Open a command prompt with administrative credentials.
2. Navigate to *System Drive:\Program Files (x86)\Microsoft System Center 2012\Orchestrator\Management Server*.
3. Type the following command: **oedc /snmp /add /ip 10.1.1.10 /port 162 /version SNMP1 /community public**
4. Restart the Orchestrator Runbook Service and the Orchestrator Runbook Server Monitor service.

Remove All SNMP Trap Destinations

You cannot remove individual SNMP trap destinations. Instead, you must remove all destinations, and then add back any that you want. To remove all SNMP trap destination, run the **oedc** command with the following syntax:

oedc /snmp /clear

▶ To remove all SNMP trap destinations

1. Open a command prompt with administrative credentials.
2. Navigate to *System Drive:\Program Files (x86)\Microsoft System Center 2012\Orchestrator\Management Server*.
3. Type the following command: **oedc /snmp /clear**
4. Restart the Orchestrator Runbook Service and the Orchestrator Runbook Server Monitor service.

Receiving SNMP Traps

After you have configured an SNMP trap destination for Orchestrator event notifications, you can receive them by using any tool that reads SNMP traps, or you can use the **Monitor SNMP Trap** activity in a runbook to read the information. The content of SNMP traps is limited to the first 1000 characters if the content exceeds that length.

The variable bindings are listed in the following table.

Trap Enterprise ID	1.3.6.1.4.1.4217.100.100
Generic ID	(6)
Specific ID	(1)
Orchestrator Event Information IDs	Orchestrator Event Type – 1 Orchestrator Event Summary – 2 Orchestrator Event Details – 3
Example	Orchestrator Event Type – 1.3.6.1.4.1.4217.100.100.1 Orchestrator Event Summary – 1.3.6.1.4.1.4217.100.100.2 Orchestrator Event Details – 1.3.6.1.4.1.4217.100.100.3

See Also

[Orchestrator Logs](#)

Audit Trail

The *Audit Trail* is a collection of text log files that contain information about the interaction of a runbook with external tools and systems. By using the Audit Trail, you can report on configuration and change compliance of processes and identify changes made to a non-Microsoft system for audit purposes or to remediate a change that causes service interruption.

Depending on how many runbooks you invoke and how many activities those runbooks contain, the Audit Trail can consume a large amount of disk space on the computer that runs the management server and runbook server. If you enable auditing, you should implement an archiving procedure to move the files generated by the Audit Trail to another computer on a regular basis.

Activating and Deactivating the Audit Trail

By default, the Audit Trail is not activated when you install Orchestrator. You can use the following procedure to activate it.

▶ To activate or deactivate the Audit Trail

1. Open a command prompt with administrative credentials.
2. Navigate to *System Drive*:\Program Files (x86)\Microsoft System Center 2012\Orchestrator\Management Server.
3. To activate the Audit Trail, type **atlc /enable**, and to deactivate the Audit Trail, type **atlc /disable**.

Audit Trail Files

Audit Trail files are stored in comma-separated value file (.csv) format. The following table shows the details.

Log Type	File name	Contents	Computer	Location
Runbook Publisher	<i>Computer Name</i> _RunbookPublisher_ <i>Timestamp</i> .csv	• Date and time that	Management Server	<i>System Drive</i> :\ProgramData\Microsoft System Center 2012\Orchestrator\Audit\ManagementService
			Runbook Server	<i>System Drive</i> :\ProgramData\Microsoft System Center

Log Type	File name	Contents	Computer	Location
		<p>t the run bo ok wa s sta rte d</p> <ul style="list-style-type: none"> • <p>Us er na me an d do ma in tha t sta rte d the run bo ok</p> <ul style="list-style-type: none"> • <p>Na me of the co mp ute r wh ere the</p>		2012\Orchestrator\Audit\RunbookService

Log Type	File name	Contents	Computer	Location
		run book ran		
Activity Runtime Information	<i>Computer Name_</i> <i>ObjectRuntimeInfo_</i> <i>Timestamp.csv</i>	<ul style="list-style-type: none"> • Date and time that activity ran • Name of runbook server that ran the activity • ID of the process that 	Runbook Server	<i>System Drive:\ProgramData\Microsoft System Center 2012\Orchestrator\Audit\PolicyModule</i>

Log Type	File name	Contents	Computer	Location
		t ran the act ivit y • Ob jec t XM L co de tha t act ivit y rec eiv ed as inp ut dat a		

When a file reaches 200 megabytes (MB) in size, a new file is created. The time stamp is included in the file name to ensure that each file name is unique. Passwords and other encrypted text fields are represented by five asterisks (*****) in the Audit Trail files.



Note

The ProgramData folder holding the audit files is often a hidden system folder.

See Also

[Orchestrator Logs](#)

Trace Logs

Orchestrator can create trace log messages on the management server to help you identify problems in the environment. By default, trace log messages are only written when there is an exception in the Orchestrator Management Service, but you can increase this level of logging by modifying a registry setting. Trace log files are available in directories for each Orchestrator feature under **C:\ProgramData\Microsoft System Center 2012\Orchestrator**. You can change these log locations by changing the registry values for the different features.



Note

The **C:\ProgramData** directory is often a hidden system folder.

Modifying Trace Log Settings

Trace log settings are configured with registry values under the key **HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\SystemCenter2012\Orchestrator\TraceLogger**. Each of the Orchestrator services has a set of registry values as shown in the following table.

Feature	Registry key
Audit Trail tool	Atlc.exe
Data Store Configuration Utility	DBSetup.exe
Management Service	ManagementService.exe
Permissions Configuration tool	PermissionsConfig.exe
Runbooks and activities	PolicyModule.exe
Runbook Designer	RunbookDesigner.exe
Runbook Server Monitor	RunbookServerMonitorService.exe
Runbook Service	RunbookService.exe
Runbook Tester	Runbook Tester.exe

The following table shows the values that are under each key. Modify these values as appropriate to change the logging details for that feature.

Registry value name	Default value	Description
LogFolder	C:\ProgramData\Microsoft System Center 2012\Orchestrator\ManagementService.exe\Logs	Location where trace logs are stored.
LogLevel	1	Level of detail of

Registry value name	Default value	Description
		information that is logged. Possible values are in the table below.
LogPrefix	ManagementService.exe	Prefix of the log file name. This prefix is followed by the date and time when the log file is created.
NewLogEvery	3600	Number of seconds until a new log file is created.

The following table provides the possible values that you can use for the log level.

Log level	Description
1	Exception detail only. This is the default setting.
3	Exception detail with warnings and errors.
7	Full logging.

See Also

[Orchestrator Logs](#)

Runbook Server Properties

The properties for a runbook server include an optional description and the account information to use for the Runbook Service. You can modify the description but can only view the service credentials. For more information about the service account and how to change it, see the **Orchestrator Security Planning** topic in **Deploying System Center 2012 - Orchestrator Release Candidate**.

To view runbook server properties

1. In the **Connections** pane, select the Runbook Servers folder. In the right pane, right-click the runbook server to select **Properties**.
2. If you want to add or change the **Description** box, type a description for this runbook server, and then click **Finish**.

See Also

[Administering System Center 2012 - Orchestrator](#)

How to Back up Orchestrator

A complete backup of an Orchestrator environment consists of the following:

- Backup of the Orchestrator database.
- File backup of the Orchestrator management server.
- File backup of each Runbook server and Orchestrator web server.

System Center 2012 - Orchestrator supports Volume Shadow copy Service (VSS) for backup and restore with System Center 2012 – Data Protection Manager (DPM). VSS is a framework that allows volume backups to be performed while an application continues to run.

Registering Orchestrator with VSS

The **SCOExpressWriter** command-line utility registers an Orchestrator database as a component associated with the Orchestrator management server. This association instructs DPM to back up the Orchestrator database when it performs a backup of the management server. Without this registration DPM must perform an individual backup of each component.

You must run **SCOExpressWriter** on the management server being registered, and you must be logged on with a user account that is a member of the local Administrators group.

The usage of this command-line utility is as follows:

```
SCOExpressWriter {/register | /unregister}
```

To register the Orchestrator database used by the local management server, run the following command:

```
SCOExpressWriter /register
```

Orchestrator Servers

Orchestrator management server, Runbook servers, and web servers do not persist any data. Runbooks and their settings are stored entirely in the Orchestrator database and accessed by these servers as required. Management servers and Runbook servers have a settings.dat file that includes configuration details to connect to the Orchestrator database. Orchestrator web servers have a web.config file with this same information. These files are backed up with standard file backups which are supported by DPM.

Orchestrator Database

The Orchestrator database is a standard SQL Server database that is supported by DPM. You should make sure to backup the service master key and store it in a secure off-site location. For more information see [BACKUP SERVICE MASTER KEY \(Transact-SQL\)](#).

Migrate Orchestrator Between Environments

This topic describes how to automatically move Orchestrator between environments. This could be useful when you want to just move to a new SQL Server 2008 R2 or if you want to move some or all of the components of Orchestrator.

The following processes and scripts enable you to easily move between environments. They are based on a full migration of all System Center 2012 - Orchestrator components to a new SQL Server 2008 R2 with a restored Orchestrator database.

The following steps are required to enable an automatic migration of Orchestrator to a new environment:

1. Backup SQL Server service master key in environment A
2. Backup the Orchestrator database in environment A
3. Deploy SQL Server 2008 R2 in environment B
4. Restore SQL Server service master key in environment B
5. Restore Orchestrator database in environment B
6. Deploy Orchestrator components in environment B



Note

See <http://go.microsoft.com/fwlink/?LinkId=246817> for information on using the Sqlcmd utility.

Back up SQL Server service master key in environment A

Back up the SQL Server 2008 R2 service master key using the following procedure as described in <http://go.microsoft.com/fwlink/?LinkId=243093>. This is a one-time operation.

Create a batch script with the following command:

```
Sqlcmd -Q"BACKUP SERVICE MASTER KEY TO FILE ='C:\BACKUP\MASTER_KEY.BAK' ENCRYPTION BY  
PASSWORD = 'password'"
```

Where 'password' is the password that will be used to protect the service master key in the file that is created. If the password is lost, the service master key cannot be recovered from the file.

Back up the Orchestrator database in environment A

Back up the entire Orchestrator database. You can perform the backup when the system is running; however it is best to perform the backup when all runbook authors have checked in any pending changes to their runbooks. Pending changes are cached on the Runbook Designer and are not backed up with a database backup.

► To back up the Orchestrator database

1. In SQL Server Management, right-click the Orchestrator database, click **Tasks**, and then click **Back up**.

2. Configure the backup settings as required in your organization.
3. Click **Script**, and then click **Script Action to New Query Window**.
4. Click **Execute** to test the backup script.
5. Create a batch file with this script. Your batch file will be similar to the following:

```
Sqlcmd -Q "BACKUP DATABASE Orchestrator TO  
DISK=N'C:\BACKUP\OrchestratorDB.bak'"
```

Deploy SQL Server 2008 R2 in environment B

Deploy SQL Server to environment B. See <http://go.microsoft.com/fwlink/?LinkID=246815> for information about creating a Sysprep image of SQL Server 2008 R2.

Restore the SQL Server service master key in environment B

Restore the Microsoft SQL Server 2008 R2 service master key by using the procedure described at <http://go.microsoft.com/fwlink/?LinkID=243093>. This will enable decryption of Orchestrator data on the new SQL server.

Create a batch script with the following command:

```
Sqlcmd -Q "RESTORE SERVICE MASTER KEY FROM FILE = 'C:\BACKUP\MASTER_KEY.BAK' DECRYPTION  
BY PASSWORD = 'password';"
```

Restore the Orchestrator database in environment B

Use the following steps to create a batch script to run on the new SQL Server computer to restore the Orchestrator database.

To create the batch file

1. In SQL Server Management, right-click the Orchestrator database, click **Tasks**, and then click **Restore**.
2. Configure the restore settings as required in your organization.
3. Click **Script**, and then click **Script Action to New Query Window**.
4. Click **Execute** to test the restore script.
5. Create a batch file with this script. Your batch file will be similar to the following:

```
Sqlcmd -Q"RESTORE DATABASE [Orchestrator] FROM DISK =  
N'C:\BACKUP\OrchestratorDB.bak'WITH FILE = 1, NOUNLOAD,  
STATS = 10"
```

Deploy Orchestrator components in environment B

Deploy Orchestrator components (management server, Web features, runbook servers, and Runbook Designers) using the silent install commands of Orchestrator setup. See **Install with the Orchestrator Command Line Install Tool** for more information on deploying Orchestrator through the command line.

The following example installs all of Orchestrator on a computer with SQL Server 2008 R2 and .NET Framework 4:

```
%systemdrive%\sco\setup\setup.exe /Silent /ServiceUserName:%computename%\administrator
/ServicePassword:password /Components:All /DbServer:%computename% /DbPort:1433
/DbNameNew:OrchestratorSysPrep /WebConsolePort:82 /WebServicePort:81 /OrchestratorRemote
/UseMicrosoftUpdate:1 /SendCEIPReports:1 /EnableErrorReporting:always
```

Sample migration scripts and commands

Backup SQL Server master service key sample

```
Sqlcmd -Q "BACKUP SERVICE MASTER KEY TO FILE ='C:\BACKUP\MASTER_KEY.BAK' ENCRYPTION BY
PASSWORD = 'password'"
```

Backup Orchestrator database sample

```
Sqlcmd -Q "BACKUP DATABASE Orchestrator TO DISK=N'C:\BACKUP\OrchestratorDB.bak'"
```

Restore SQL Server master service key sample

```
Sqlcmd -Q "RESTORE SERVICE MASTER KEY FROM FILE =
'c:\temp_backups\keys\service_master_key' DECRYPTION BY PASSWORD = 'password'"
```

Restore Orchestrator database sample

```
Sqlcmd -Q "RESTORE DATABASE [Orchestrator] FROM DISK =
N'C:\BACKUP\OrchestratorDB.bak'WITH FILE = 1, NOUNLOAD, STATS = 10"
```

Install Orchestrator from batch file sample

```
%systemdrive%\sco\setup\setup.exe /Silent /ServiceUserName:%computename%\administrator
/ServicePassword:password /Components:All /DbServer:%computename% /DbPort:1433
/DbNameNew:OrchestratorSysPrep /WebConsolePort:82 /WebServicePort:81 /OrchestratorRemote
/UseMicrosoftUpdate:1 /SendCEIPReports:1 /EnableErrorReporting:always
```

Best Practices Analyzer

The Microsoft System Center 2012 - Orchestrator Best Practices Analyzer is a tool that looks at the configuration data in an Orchestrator deployment and identifies settings that may cause issues within your environment. It performs the following functions:

- Gathers information about an Orchestrator deployment
- Determines if the configurations are set according to the Microsoft recommended best practices
- Reports on collected configurations, indicating settings that differ from recommendations
- Indicates potential problems in the deployment

Installation

The Orchestrator BPA must be installed on the Orchestrator Management server. It depends on having the Microsoft Baseline Configuration Analyzer 2.1 (<http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=16475>) installed on the Orchestrator Management Server prior to installation.

Install process:

1. Install Microsoft Baseline Configuration Analyzer 2.1
2. Run Microsoft.SystemCenter.2012.Orchestrator.BestPracticesAnalyzer.msi

Uninstall process:

1. Remove Microsoft System Center 2012 Orchestrator – Best Practices Analyzer (BPA) from the Programs and Features panel within Windows Control Panel.

Usage

1. Run the Microsoft Baseline Configuration Analyzer 2.1 from the Start menu in windows.



Note

To learn how to run the Microsoft Baseline Configuration Analyzer scans from PowerShell cmdlets, please see the MBCA 2.1 help file.

2. Select System Center 2012 – Orchestrator BPA from the product dropdown within the MBCA 2.1 user interface.
3. Click Start Scan.
4. Review the results and recommendations.

Orchestrator rules

System Center 2012 - Orchestrator configuration checks:

Orchestrator Component	Rule	Description
Management Server	Check log purge is set to the default value	This rule checks that the default log purging for Runbooks is set to the default values for running every one day and to keep the last five hundred entries. Please refer to http://go.microsoft.com/fwlink/?LinkID=239613

Orchestrator Component	Rule	Description
		for more information about setting the purging policy for Runbook logs.
Management Server	Check the refresh interval for permissions in the Orchestration console	This rule checks that the default refresh interval for generating the cache that allows access to runbooks from the Orchestration Console is set to six hundred seconds. Please refer to http://go.microsoft.com/fwlink/?LinkID=239946 for more information on how the refresh cache is set up.
Management Server	Check logging on ManagementService.exe	This rule checks that the default trace logging on the ManagementService.exe is set to the default of 1. This can impact performance if a different value is configured. Please refer to http://go.microsoft.com/fwlink/?LinkID=239530 for more information on configuring trace logs.
Management Server	Check logging on PermissionsConfig.exe	This rule checks that the default trace logging on the PermissionsConfig.exe is set to the default of 1. This can impact performance if a different value is configured. Please refer to http://go.microsoft.com/fwlink/?LinkID=239530 for more information on configuring trace logs.
Management Server	Check logging on Runbooks	This rule checks if logging is enabled on Runbooks. This can impact performance if logging is enabled on frequently used Runbooks. Please refer to http://go.microsoft.com/fwlink/?LinkID=239614 for more information on enabling logging on Runbooks.
Management Server	Check memory on Management Servers	This rule checks that the memory allocated to the Management Server is greater than 2048MB. If the Management Server has less than 2048MB you should monitor its performance to ensure it is meeting the expected goals in the environment. Please refer to http://go.microsoft.com/fwlink/?LinkID=242545 for information on the Orchestrator

Orchestrator Component	Rule	Description
		Management Pack.
Runbook Server	Check that maximum concurrent runbooks is set to the default value of 50.	This rule checks that the maximum number of concurrent runbooks that are configured to run on a Runbook Server is set to fifty. This can impact performance if a different value is configured. Please refer to http://go.microsoft.com/fwlink/?LinkID=239560 for more information on configuring Runbook throttling.
Runbook Server	Check logging for PolicyModule.exe	This rule checks that the default trace logging on the PolicyModule.exe is set to the default of 1. This can impact performance if a different value is configured. Please refer to http://go.microsoft.com/fwlink/?LinkID=239530 for more information on configuring trace logs.
Runbook Server	Check logging for RunbookService.exe	This rule checks that the default trace logging on the RunbookService.exe is set to the default of 1. This can impact performance if a different value is configured. Please refer to http://go.microsoft.com/fwlink/?LinkID=239530 for more information on configuring trace logs.
Runbook Server	Check memory on Runbook Servers	This rule checks that the memory allocated to the Runbook Servers is greater than 2048MB. If the Runbook Server has less than 2048MB you should monitor its performance to ensure it is meeting the expected goals in the environment.
Web components	Check memory on the server hosting the Orchestration Console	This rule checks that the memory allocated to the Orchestration Console server is greater than 2048MB. If the server has less than 2048MB you should monitor its performance to ensure it is meeting the expected goals in the environment. Please refer to http://go.microsoft.com/fwlink/?LinkID=242545 for information on the Orchestrator Management Pack.
Web components	Check users group	This rule checks that the windows group that is used to manage access to runbooks is

Orchestrator Component	Rule	Description
	managing runbook access	configured as a domain group if the web components are not installed on the Management Server. The group must be a domain group in order for users to have access through the web service and Orchestration console when the web components are installed on a separate server than the Management Server. Please refer to http://go.microsoft.com/fwlink/?LinkID=239561 for more information on configuring the Orchestrator users group.
Designer	Check memory on the computer hosting the Orchestrator Designer	This rule checks that the memory allocated to the Orchestrator Designer is greater than 2048MB. If the computer has less than 2048MB you should monitor its performance to ensure it is meeting the expected goals in the environment. Please refer to http://go.microsoft.com/fwlink/?LinkID=242545 for information on the Orchestrator Management Pack.

How to Benchmark

Orchestrator runbook activities can be thought of as having two distinct types of code: platform code and domain code. The term *domain code* is used to identify code within a runbook activity that is typically not associated with the Orchestrator platform itself (with notable exceptions, such as **Invoke Runbook**, **Junction**, and others). For example, the **Invoke Web Service** standard activity would contain Orchestrator platform code (the “plumbing” of the activity) as well as domain code unique to invoking a SOAP-based web service. The platform code will be very similar for most activities, since it is built on a common framework. However, there will potentially be great variation in domain code for different activities.

Data Logging

Another aspect of runbook performance is data logging. For the purpose of understanding performance consider two logging configurations: Default logging and Common Published Data logging. Default logging results in approximately 524 bytes of data being written to the Orchestrator database each time an activity is run. Logging of common published data writes approximately 6,082 bytes of data (12 times the default logging level). There is a notable difference in performance between these logging levels.

Consider the scenario where the same runbook activity is run twice, once with data logging at the default level and once with logging of common published data enabled. The domain code should take the same amount of time to complete. However, the platform code will take longer to run with common published data logging enabled. Essentially, the platform code has to support logging 12 times more data with common published data enabled than it did when running at the default logging level.

The Standard Activity **Compare Values** can be used to create benchmarks of an Orchestrator environment.

► **To create a runbook that can be used to benchmark your Orchestrator environment**

1. Create a new runbook.
2. Add a **Compare Values** activity from the Standard Activity palette. Double-click the activity to configure it.
3. Click the **General** tab and configure this activity to compare strings (the default value).
4. Click the **Details** tab, type the value **STRING** in the **Test** box and select **is empty**.
5. Click **Finish** to save the updates to the activity.
6. Right-click the activity and select **Looping**.
7. Select the **Enable** checkbox and enter the number **0** (zero) for **Delay between attempts**.
8. Click the **Exit** tab.
9. Change the default exit condition. Click **Compare Values**, check the **Show Common Published Data** checkbox, and select **Loop: Number of attempts**. Click **OK** to save this change.
10. Select **value** from the updated exit condition and type the number **10000** (ten-thousand). Click **OK** to save this change.
11. Click **Finish** to save these updates.
12. Click **Check In** to save the changes to the Orchestrator database.

This simple one-activity runbook will run a **Compare Values** activity 10,000 times. **Compare Values** is a very simple activity whose domain code is quite minimal. This runbook can be invoked under a variety of circumstances to characterize the overall performance of a given Orchestrator runtime environment.

This runbook can be used to experiment with different configurations of Orchestrator. For example, supposed you wanted to determine the performance of four Runbook Servers deployed to different data centers.

Data Center	Logging Configuration	Platform Code Run Time (seconds)	ms/Activity	Scale Factor
Location 1	Default logging	819	82	1.0 (reference)
Location 1	Logging common	2012	201	2.5

Data Center	Logging Configuration	Platform Code Run Time (seconds)	ms/Activity	Scale Factor
	published data			
Location 2	Default logging	1229	123	1.5
Location 2	Logging common published data	3686	369	4.5
Location 3	Default logging	2457	426	3.0
Location 3	Logging common published data	4422	442	5.4
Location 4	Default logging	1474	147	1.8
Location 4	Logging common published data	2654	265	3.2

Notice the significant decrease in platform performance caused by logging of common published data. The worst scenario appears to be logging of common published data at Location 2. On the surface, this appears to be a clear and relevant conclusion.

However, it should be noted that these figures reflect the overhead of the platform code, not the domain code. Domain code runtimes can be significantly longer. For example, the **Create VM from Template** activity in the Virtual Machine Manager Integration Pack may run for several minutes as the VM is created. Expanding on the previous example, consider the platform code costs on a runbook activity that takes 1 minute to run (1 minute = 60,000 milliseconds) regardless of location.

Data Center	Logging Configuration	Platform Code Run Time (seconds)	% Domain Code	% Platform Code
Location 1	Default logging	819	98.6%	1.4%
Location 1	Logging common published data	2012	96.7%	3.3%
Location 2	Default logging	1229	98.0%	2.0%
Location 2	Logging common published data	3686	93.9%	6.1%
Location 3	Default logging	2457	95.9%	4.1%
Location 3	Logging common published data	4422	92.6%	7.4%

Data Center	Logging Configuration	Platform Code Run Time (seconds)	% Domain Code	% Platform Code
Location 4	Default logging	1474	97.5%	2.5%
Location 4	Logging common published data	2654	95.6%	4.4%

A clearer picture begins to emerge from the data. The scenario where logging of common published data is enabled at Location 2 continues to be the worst performer. However, the platform code and logging only accounts for 6% of the total runtime. While this is a significant figure, the best-case scenario is 1.4%. Essentially, the time spent in the domain code in the example far outweighs the time spent running platform code. To put this in perspective, if you were able to completely eliminate the platform code costs, you would only see runbook performance improvements in the range of 1.4 to 7.4%.

Of course most real-world scenarios will be different. Activity behavior may change depending on what the domain code is told to do. For example, a **Clone VM from Template** activity may take one minute to clone a VM from Server Template A, but take 5 minutes to clone a VM from Server Template B. Also, Runbook Servers may reside on different networks with different performance characteristics which can potentially impact both domain code performance as well as Orchestrator data logging performance.

To summarize:

- Make careful decisions about when to log published data.
- Carefully consider the impact of logging common published data. Remember that the number of times activities run determines the volume of logged data. A runbook with a small number of activities run many times can result in more data logging than a larger runbook run a small number of times.
- Do not enable logging of activity specific published data in production environments.
- Develop an understanding of how much time your runbooks spend running domain code compared to running platform code.
- Estimate platform code costs using the techniques outlined in this document. Use as a reference in considering where to make improvements in runbook performance.
- Use the techniques outlined in this document to gain a deeper understanding of the relative performance of your different runtime environments. Identify opportunities for improvement by making normalized comparisons of your measurements.

How to Recover a Database

The Orchestrator database can be backed up and restored using most standard MS SQL Server database backup/restore mechanisms. This includes Microsoft SQL Server Backup, DPM SQL Server backup, and others. Orchestrator provides a VSS Writer that will discover the database

server that is associated with the Management Server and back up the database when the Management Server is backed up.

However, there are a few key considerations when restoring.

Orchestrator Cryptography

System Center 2012 Orchestrator provides a set of services for encryption and decryption of runbook properties and published data. These services are based on Microsoft SQL Server 2008 R2 cell-level encryption. The Orchestrator database has a database encryption key that is created during its installation. This key is generated using a random passphrase. When a full database backup is performed, the key is backed up with the database. Likewise, the key is restored when the database is restored.

However, the encryption services also depend on the MS SQL Server Service Master Key. The service master key should be backed up and stored in a secure, off-site location. Creating this backup should be one of the first administrative actions performed on the server. The procedure for doing this is documented for [Microsoft SQL Server 2012](http://go.microsoft.com/fwlink/?LinkId=249148) (<http://go.microsoft.com/fwlink/?LinkId=249148>).

The database key is essentially paired with the service master key on the database server targeted by the installer. If either the database key or the service master key is lost, encrypted data stored in the data is likewise lost. This would include the license key, either entered by the user or an automatically created trial license.

► To perform a backup

1. Back up the Microsoft SQL Server service master key using the procedure for [backing up the service master key for Microsoft SQL Server 2012](#). This is a one-time operation. Note "password" is the password that will be used to protect the service master key in the file that is created. If the password is lost, the service master key cannot be recovered from the file.

```
BACKUP SERVICE MASTER KEY TO FILE = 'path_to_file'  
ENCRYPTION BY PASSWORD = 'password'
```

2. Back up the entire Orchestrator database. The backup may be performed when the system is running, but it is best to perform the backup when all runbook authors have checked in any pending changes to their runbooks. Pending changes are cached on the Runbook Designer and are not backed up with a database backup.

► To restore the database

1. If you are restoring to the same database server from which the backup was taken, and the service master key has not changed, simply restore the backup.
2. If you are restoring to a different database server with a different service master key, or you are restoring to the same database from which the backup was taken but the service master key has changed, the service master key must be restored to match the one used during the database backup. Use the procedure for [restoring the service master key for](#)

[Microsoft SQL Server 2012](http://go.microsoft.com/fwlink/?LinkId=249149) (http://go.microsoft.com/fwlink/?LinkId=249149).

```
BACKUP SERVICE MASTER KEY TO FILE =  
'c:\temp_backups\keys\service_master_key'  
ENCRYPTION BY PASSWORD = '3dH85Hhk003GHk2597jheij4'
```



Note

If there are multiple databases using this service master key for encryption on your Microsoft SQL Server, all of these databases could be affected by this change. Consulting with your DBA before performing this administrative task is strongly recommended.

3. Restore the database from the backup.
4. On the Orchestrator Management Server, run the Data Store Configuration utility from the Start menu.
5. Provide the connection details to connect to the new database. Note: Do not use "localhost" or ".". Explicitly specify the database server name and database name.
6. Restart the Management Service.
7. Run the Data Store Configuration utility on each Runbook Server. This utility is not located in the Start menu on Runbook Servers. It can be found in <SC2012OrchestratorInstallDir>\Microsoft System Center 2012\Orchestrator\Runbook Server. Note: For Runbook Servers installed on the same server as the Management Server one doesn't need to run the Data Store Configuration utility a second time. Running it once will update the configuration for both the Management Server and Runbook Server at the same time.
8. Restart the Runbook Server(s).
9. Follow the Web Components Recovery Process to update the Web Components to connect to the new database.

Database Sizing and Performance

Database sizing is the key to understanding the performance of System Center 2012 - Orchestrator. The runbook servers, management server, and web components all depend on the Orchestrator database for their operations. Problems with Orchestrator deployments can arise from an incomplete understanding of the types of data in the database and how to manage them. Because the Runbook Designer communicates with the Orchestrator database (through the management server), poor database performance will impede that communication.

The Orchestrator operator experience is based on two components: The **Orchestration Console** and the Web Service. The **Orchestration Console** is a Silverlight-based application that depends on the Web Service for its connection to the Orchestrator database. The Web Service is an IIS application that connects to the database. Consequently, the Web Service and **Orchestration Console** are both dependent on the performance of the Orchestrator database.

Additionally, while the **Orchestration Console** is dependent on the Web Service, it also has logic unique to its function as a user interface and its own performance characteristics.

Key Concepts

Configuration Data and Log Data

At a high level the Orchestrator database contains two kinds of data:

- Configuration Data

The Orchestrator infrastructure contains configuration data. This data is not a concern in the context of database growth because the storage requirements for this type of data are small.

- Log Data

Orchestrator creates different types of log data, all of which can be viewed and managed in the **Runbook Designer**. The storage requirements for this data can vary in size and be large.

The following table lists the types of log data that can be stored in the Orchestrator database. Orchestrator also stores data in separate log files (outside of the database) for audit trails and tracing. For more information about all the types of log data, see [Orchestrator Logs](#).

Type of Log Data	Location in Runbook Designer	Managed by Log Purge?
Runbook logs	Log and Log History tabs	Yes
Activity (Platform) events	Events tab	Yes
Audit history	Audit History tab	No

Platform Code and Domain Code

Orchestrator runbook activities contains two distinct types of code:

- Platform Code

This is common code shared by most activities, and is used to run common tasks performed by Orchestrator activities. Platform code generates Common Published Data.

- Domain Code

Runs a variety of tasks that are specific for the actions for each activity, that are typically not associated with the Orchestrator platform itself. Potentially, there can be great variation between platform code and domain code.

The logging data generated for a given activity can contain data elements that are single or multi-valued. Every activity produces a single record of single-value data. Domain code can produce multiple records of multi-value data and is therefore responsible for determining what the activity does with the common published data it has received from prior activities.

Essentially, Orchestrator runbooks are designed to pass data between discrete elements of domain code. Also, domain code can optionally generate Activity-specific Published Data.

All runbooks have core similarity in that they run activities that consist of domain code and platform code, they loop workflows and they branch. Branching is when a runbook calls other runbooks to do a specific task. When a runbook is first invoked it consists of a single thread. When this thread encounters a runbook activity whose links require a branch, additional threads are created, one for each branch. Each thread takes as input the common published data from the activity that created the branch. This data is correlated back to the prior activities in the runbook to update the common published data that the activities subscribe to.

Domain code potentially affects database performance more than multi-threading generated by branching. This is because domain code can potentially generate large amounts of activity-specific published data.

Logging Options

The **Logging** tab on the **Properties** for a runbook allows you to optionally store logging entries. The term *default logging* refers to having neither of the two published data options selected, which amounts to 524 bytes generated for each activity. The logging options provide for two categories of common published data:

- *Common Published Data*

The set of data items common to all activities. For a list, see the [Runbook Log Options](#) section in [Runbook logs](#).

This logging option generates 6082 bytes for each activity.

- *Activity-specific Published Data*

The set of data that is specific to the activity that is optionally created by domain code.

This logging option generates 6082 bytes in addition to the bytes logged by specific activities.



Tip

This option is selected primarily for debugging purposes. Leave unchecked to limit logging growth.

Setting logging options can significantly affect performance and increase database growth. Consider the scenario where the same runbook activity is run twice, first with data logging at the default level (no published data options selected) and then set with common published data selected. The domain code should take the same amount of time to complete. However, the platform code will take longer to run because it has to support 12 times the amount of common published data logging than it does with just default logging.

Purging Logs

The default options specified for the **Log Purge** feature in the **Runbook Designer** is configured to provide the best user experience for an out-of-the-box Orchestrator deployment. Changing these values can change the performance characteristics of the environment, and should be implemented gradually and high-watermarked, so that the impact of the change can be evaluated.

For more information on automatic and manual purging of logs, see the [Purging Runbook Logs](#) section of [Runbook logs](#).

Creating Performance Benchmarks

To create a simple runbook to test logging growth you can use the Standard Activity **Compare Values** to create benchmarks of an Orchestrator environment.

The following procedure creates a simple runbook that runs a **Compare Values** activity 10,000 times. **Compare Values** is a very simple activity whose domain code is quite minimal. This runbook can be invoked under a variety of circumstances to characterize the overall performance of a given Orchestrator runtime environment.

▶ To create a runbook that can be used to benchmark your Orchestrator environment

1. Create a new runbook.
2. Add a **Compare Values** activity from the Standard Activity palette. Double-click the activity to configure it.
3. Click the **General** tab and configure this activity to compare strings (the default value).
4. Click the **Details** tab, type the value **STRING** in the **Test** box and select **is empty**.
5. Click **Finish** to save the updates to the activity.
6. Right-click the activity and select **Looping**.
7. Select the **Enable** checkbox and enter the number **0** (zero) for **Delay between attempts**.
8. Click the **Exit** tab.
9. Change the default exit condition. Click **Compare Values**, check the **Show Common Published Data** checkbox, and select **Loop: Number of attempts**. Click **OK** to save this change.
10. Select **value** from the updated exit condition and type the number **10000** (ten-thousand). Click **OK** to save this change.
11. Click **Finish** to save these updates.
12. Click **Check In** to save the changes to the Orchestrator database.

This runbook can be used to experiment with different configurations of Orchestrator. For example, you can create the benchmark runbooks to determine the performance of four Runbook Servers deployed to different data centers.

Data Center	Logging Configuration	Platform Code Run Time (milliseconds)	Milliseconds per Activity	Scale Factor
Location 1	Default logging	819	82	1.0 (reference)
Location 1	Logging common published data	2012	201	2.5
Location 2	Default logging	1229	123	1.5

Data Center	Logging Configuration	Platform Code Run Time (milliseconds)	Milliseconds per Activity	Scale Factor
Location 2	Logging common published data	3686	369	4.5
Location 3	Default logging	2457	426	3.0
Location 3	Logging common published data	4422	442	5.4
Location 4	Default logging	1474	147	1.8
Location 4	Logging common published data	2654	265	3.2

Notice the significant decrease in platform performance caused by logging of common published data. The worst scenario appears to be logging of common published data at Location 2. On the surface, this appears to be a clear and relevant conclusion.

However, it should be noted that these figures reflect the overhead of the platform code, not the domain code. Domain code runtimes can be significantly longer. For example, the **Create VM from Template** activity in the Virtual Machine Manager Integration Pack may run for several minutes as the VM is created. Expanding on the previous example, consider the platform code costs on a runbook activity that takes 1 minute to run (1 minute = 60,000 milliseconds) regardless of location.

Data Center	Logging Configuration	Platform Code Run Time (milliseconds)	% Domain Code	% Platform Code
Location 1	Default logging	819	98.6%	1.4%
Location 1	Logging common published data	2012	96.7%	3.3%
Location 2	Default logging	1229	98.0%	2.0%
Location 2	Logging common published data	3686	93.9%	6.1%
Location 3	Default logging	2457	95.9%	4.1%
Location 3	Logging common published data	4422	92.6%	7.4%
Location 4	Default logging	1474	97.5%	2.5%
Location 4	Logging common published data	2654	95.6%	4.4%

A clearer picture begins to emerge from the data. The scenario where logging of common published data is enabled at Location 2 continues to be the worst performer. However, the platform code and logging only accounts for 6% of the total runtime. While this is a significant figure, the best-case scenario is 1.4%. Essentially, the time spent in the domain code in the example far outweighs the time spent running platform code. To put this in perspective, if you were able to completely eliminate the platform code costs, you would only see runbook performance improvements in the range of 1.4 to 7.4%.

Of course most real-world scenarios will be different. Activity behavior may change depending on what the domain code is told to do. For example, a **Clone VM from Template** activity may take one minute to clone a VM from Server Template A, but take 5 minutes to clone a VM from Server Template B. Also, Runbook Servers may reside on different networks with different performance characteristics which can potentially impact both domain code performance as well as Orchestrator data logging performance.

Determining Database Growth

Your database administrator for the Orchestrator database can use the following guidelines for determining database file growth strategy:

- In general, the database files will not increase in size with each invocation of a runbook. The files will grow when the data contained within them reaches a certain high watermark configured by your database administrator, at which time the file will generally be expanded.
- Each time a runbook activity runs it should be counted individually, which should be considered when looping features can cause a single activity to run multiple times.
- To determine the storage needed for each invocation of the runbook, multiply the number of activities in the runbook by the number of bytes added to the database according the selected logging level. These values are as follows:
 - 524 bytes
Default logging configuration.
 - 6082 bytes
Common published data logging level.
 - 6082 bytes + data logged by activity
Activity-specific published data logging level.
- By default, Orchestrator purges all but the most recent 500 logs for each runbook nightly at 2:00 am. To determine the storage required for each invocation of the runbook, multiply the storage needed for each invocation of the runbook by 500. If you change the Log Purge setting, multiply each invocation by the estimated number of invocations per day, week, or month as needed.

The following table shows growth and performance estimates for the logging level configurations.

Logging Level	DB Growth Factor	Performance Factor	Recommended for Production
Default	1	1	Yes
Common published data	11.6x	2.5x	Limited use with planning
Activity-specific published data	Greater than 11.6x	Greater than 2.5x	No

Examples

Example 1

The following table describes the database sizing considerations for a deployment of Orchestrator.

Runbook Name	Number of Activities	Logging Level	Invocations per Day
Runbook 1	50	Default	100
Runbook 2	25	Default	50
Runbook 3	12	Common published data	24
Runbook 4	8	Common published data	500

Using the database sizing described above, you can estimate the storage requirements for the runbooks.

Runbook Name	Bytes per Invocation	Storage in MB Default Log Purge (500 invocations)	Invocations per Month	Storage in MB One Month (Not Default Log Purge)	% of DB storage after 30 Days
Runbook 1	26,200	12.5	3,000	74.5	9%
Runbook 2	13,100	6.2	1,500	18.7	2%
Runbook 3	72,984	34.8	720	50.1	6%

Runbook Name	Bytes per Invocation	Storage in MB Default Log Purge (500 invocations)	Invocations per Month	Storage in MB One Month (Not Default Log Purge)	% of DB storage after 30 Days
Runbook 4	48,656	23.2	15,000	696.0	83%
		Total: 76.7 MB		Total: 839.8 MB	

This example clearly illustrates the importance of making sound decisions for data logging. Runbook 4 contains only eight activities, but when configured at the Common Published Data Logging level, it consumes most of storage in the database because of the high frequency of invocation. Based on these results you may prefer to reduce the logging level of Runbook 4 to the Default logging configuration.

Example 2

The following table describes the database sizing considerations for another deployment of Orchestrator.

Runbook Name	Number of Activities	Logging Level	Invocations per Day
Runbook 1	50	Default	100
Runbook 2	25	Default	50
Runbook 3	12	Common published data	24
Runbook 4	8	Default	500

Recalculating the storage figures for the updated configuration produces significantly different results.

Runbook Name	Bytes per Invocation	Storage in MB Default Log Purge (500 invocations)	Invocations per Month	Storage in MB One Month (Not Default Log Purge)	% of DB storage after 30 Days
Runbook 1	26,200	12.5	3,000	74.5	37%
Runbook 2	13,100	6.2	1,500	18.7	9%

Runbook Name	Bytes per Invocation	Storage in MB Default Log Purge (500 invocations)	Invocations per Month	Storage in MB One Month (Not Default Log Purge)	% of DB storage after 30 Days
Runbook 3	72,984	34.8	720	50.1	25%
Runbook 4	4,192	2.0	15,000	60.0	29%
		Total: 55.5 MB		Total: 203.8 MB	

While there is very little change in the default logging configuration (500 log entries per runbook), the 30-day storage requirements have changed greatly. Clearly the storage cost of using Common Published Data logging for Runbook 4 should be carefully considered since this change results in a 76% reduction in database storage requirements for 30 days of data.

Summary

Use the following guidelines to manage database sizing and performance:

- Enable logging of Common Published Data only if needed.
- Remember that the number of times activities run determines the volume of logged data. A small runbook with a few of activities run several times can result in more data logging than a larger runbook run a fewer number of times.
- Do not enable logging of Activity-specific Published Data in production environments, and should only be used for debugging purposes.
- Develop an understanding of how much time your runbooks spend running domain code compared to running platform code.
- Estimate platform code costs using the techniques outlined in this document. Use as a reference in considering where to make improvements in runbook performance.
- Identify opportunities for improvement by making normalized comparisons of your measurements.

See Also

[Orchestrator Logs](#)

[Runbook logs](#)

Orchestrator Architecture

How to Recover Web Components

The Web Service database reference does not get modified by the Database Configuration Utility (only the installer performs this task). You will need to manually modify it.

Web Components Recovery Process

To do this, you will need to complete the following actions:

► To modify the Web Service database reference

1. Open a Command Prompt using **Run as administrator**.
2. Execute the following command (assuming the default installation path):

```
C:\Windows\Microsoft.NET\Framework\v4.0.30319\aspnet_regiis.exe -pdf "connectionStrings" "C:\Program Files (x86)\Microsoft System Center 2012\Orchestrator\Web Service\Orchestrator2012"
```
3. Open IIS Manager and navigate to the Orchestrator2012 virtual application.
4. Open up Connection Strings and then modify OrchestratorContext. Locate the segment that starts with “provider=System.Data.SqlClient;provider connection string” and then modify the Data Source and Initial Catalog attributes according to your new SQL Server and Database Catalog name respectively, then click OK.
5. If you want to re-encrypt the connection strings, you can execute the following command at the command prompt:

```
C:\Windows\Microsoft.NET\Framework\v4.0.30319\aspnet_regiis.exe -pef "connectionStrings" "C:\Program Files (x86)\Microsoft System Center 2012\Orchestrator\Web Service\Orchestrator2012"
```

Feature Performance Considerations

Despite the great deal of variance in their design and visual complexity, runbooks are essentially very simple. Runbooks all essentially do three things: Run activities, manage published data, and branch.

Orchestrator Feature Functions

Additionally, runbook activities can be thought of as having two distinct types of code: Platform code and domain code. The term *domain code* is used to identify code called within a runbook activity typically not associated with the Orchestrator product itself. For example, the **Invoke Web Service** standard activity would contain Orchestrator platform code (the “plumbing” of the activity) as well as domain code unique to invoking a web service. The platform code will be unique but similar for most activities, since it is built from a common framework. However, there will potentially be great variation in domain code for different activities.

Essentially, Orchestrator runbooks are designed to pass data between discrete elements of domain code.

While technically not mandatory, every activity generally consumes published data created by prior runbook activities. What a given activity does with published data it subscribes to is entirely up to the domain code.

All runbook activities create published data, which is referred to as *Common Published Data*. Domain code will generally create published data, generally referred to as *Activity Specific Published Data*. This data will be as unique to a given activity as the domain code itself. Also, it's not required that domain code produce published data.

The data produced by a given activity can contain data elements that are single or multi-valued. For example, every activity produces a single record of single-value data referred to as common published data. Domain code can produce multiple records of multi-value data.

Branching is a fundamental concept for Orchestrator. A given runbook activity will create a branch if it is the origin of two or more links whose filter conditions indicate there is data to pass to the activity at the end of the link. When a runbook is first invoked it consists of a single thread. When this thread encounters a runbook activity whose links require a branch, additional threads are created, one for each branch. Each thread takes as input the published data from the activity that created the branch. This data is correlated back to the prior activities in the runbook (hence the ability to subscribe to published data from prior activities).

The operator experience is based on two components: The Orchestration Console and the Web Service. The Orchestration Console is a Silverlight application that depends on the Web Service for its connection to the Orchestrator database. The Web Service is an IIS application that connects to the database. Hence the Web Service and Orchestration Console are very dependent on the performance of the Orchestrator 2012 database.

Parts of the Orchestrator 2012 database are new to the product and directly support the Web Service. However, parts of the Web Service depend on the legacy Opalis 6.3 database structure. Additionally, while the Orchestration Console is dependent on the Web Service, it also has logic unique to its function as a user interface that will have its own performance characteristics.

How to Use the Integration Toolkit to Extend Orchestrator Capabilities

The System Center 2012 Service Pack 1 (SP1) Orchestrator Integration Toolkit is a set of tools to help you create new integrations for Orchestrator. Using wizard-based applications, you can easily create new workflow activities and Integration Packs that extend the capabilities of the product. You can also create custom workflow activities using the Orchestrator SDK and C#, and then package them into an Integration Pack using this toolkit. For complete documentation for the Orchestrator Integration Toolkit, see the [MSDN Library](#).