

System Center Orchestrator 2012 Runbook Guide

Microsoft Corporation

Published: September 2011

Feedback

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

Copyright

This document is provided "as-is". Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. 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.

© 2011 Microsoft Corporation. All rights reserved.

Microsoft, Active Directory, ActiveSync, Internet Explorer, JScript, SharePoint, SQL Server, Visio, Visual Basic, Visual Studio, Win32, Windows, Windows PowerShell, Windows Server, and Windows Vista are trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.

Contents

System Center Orchestrator 2012 Runbook Guide.....	4
Runbook Concepts.....	4
Runbooks.....	5
Runbook Properties.....	5
Runbook Permissions	10
Activities.....	10
Standard Activities.....	11
Monitors.....	12
Customized Activities	12
Common Activity Properties	12
Workflow Control.....	14
Starting Point	14
Smart Links.....	15
Embedded Loops	19
Tools.....	21
Runbook Designer	21
Runbook Tester	23
Design and Build Runbooks.....	24
Designing a Runbook	25
Building a Runbook.....	25
Data Manipulation.....	27
Computer Groups.....	28
Counters.....	29
Functions.....	30
Regular Expressions	34
Schedules	36
Variables	38
Published Data	39
How to Test a Runbook	44
Troubleshooting	44
Runbook logs.....	45
Activity Events	48
Audit Trail	50
Trace Logs.....	54
Deploy and Launch Runbooks.....	56
Deploying Runbooks.....	56
Running Runbooks	57
Runbook Server Properties	58
Creating and Testing a Sample Runbook	59

System Center Orchestrator 2012 Runbook Guide

Welcome to the System Center Orchestrator 2012 Runbook Guide. This document describes Orchestrator runbooks. This document provides information on the tools available in Orchestrator and guidance on how to automate tasks and processes for your IT environment. Topics include how to write, test, and deploy a runbook with System Center Orchestrator 2012. Sample runbooks are included in a section at the end of this guide.

Orchestrator Runbooks

- [Runbook Concepts](#)
Basic information and terminology for runbooks, activities, and workflows.
- [Tools](#)
Tools to build and launch runbooks.
- [Design and Build Runbooks](#)
How to design and create a runbook.
- [Deploy and Launch Runbooks](#)
How to deploy runbooks into your Orchestrator environment.

Other resources for this product

- TechNet Library main page for **System Center Orchestrator 2012**
- **System Center Orchestrator 2012 Getting Started Guide**
- **System Center Orchestrator 2012 Deployment Guide**
- **System Center Orchestrator 2012 Integration Packs**
- **System Center Orchestrator 2012 Runbook Activity Reference**

Runbook Concepts

The power of System Center Orchestrator 2012 is in the runbook and the individual activities that make up a runbook. Runbooks contain the instructions, or workflow, for an automated task or process. The individual steps within a runbook are called activities. Within the workflow are additional controls that provide information and instructions to control the sequence of activities within the runbook. Runbooks, activities, and each of the workflow controls have configurable properties. You modify these properties in order to configure the behavior that your workflow requires.

The topics covered in this section provide detailed information on the attributes and features related to runbooks.

Runbook Concepts

- [Runbooks](#)
Configuration information for runbooks.
- [Activities](#)
Categories of activities and their attributes
- [Workflow Control](#)
Tools to manage a runbook workflow.

Other resources for this product

- TechNet Library main page for **System Center Orchestrator 2012**
- [System Center Orchestrator 2012 Runbook Guide](#)
- [Tools](#)
- [Design and Build Runbooks](#)
- [Deploy and Launch Runbooks](#)

Runbooks

Runbooks allow a wide range of customization options. This section provides details for all properties and permission settings for runbooks.

Runbook Attributes

- [Runbook Properties](#)
Configuration information for individual runbooks.
- [Runbook Permissions](#)
Access rights and permissions for single and multiple runbooks.

Other resources for this product

- TechNet Library main page for **System Center Orchestrator 2012**
- [System Center Orchestrator 2012 Runbook Guide](#)
- [Runbook Concepts](#)
- [Activities](#)
- [Workflow Control](#)

Runbook Properties

A runbook is essentially a series of activities utilizing data, performing tasks, and publishing data for use by other activities within the runbook. Each runbook has a collection of configurable properties. These properties allow you to customize the behavior of a runbook.

▶ **To view the properties of a runbook**

1. In the Runbook Designer, in the **Connections** pane, click the **Runbooks** folder.
If the runbook is stored in a folder, then select the appropriate folder under **Runbooks**.
- 2.
3. In the Runbook Designer workspace, right-click the tab of a runbook, and then select **Properties**.
4. To close the **Runbook Properties** dialog, click **Finish**.



Note

If the runbook is checked in, when you save changes to the properties, you will receive an alert confirming that you want to check out the runbook.

A summary of the runbook properties and how to configure them follows.

General

The **General** tab of the runbook properties allows you to set a custom name and description for the runbook. You can also associate a schedule with the runbook. Once assigned a schedule, the runbook will only run on the dates and times specified in the schedule.

▶ To create a schedule

1. In the Runbook Designer, in the **Connections** pane, expand the **Global Settings** folder.
2. Right-click the **Schedules** folder, select **New**, and then select **Schedule**.
3. On the **General** tab of the **New Schedule** dialog, provide a **Name** for the schedule.
4. On the **Details** tab of the **New Schedule** dialog, choose the date and time period to launch the runbook.

▶ To configure the schedule for specific days of the week

- a. On the **Details** tab of the **New Schedule** dialog, select the **Days of the week** radio button, and select the days you want to launch the runbook.
- b. Under **Occurrence**, select the week of the month to launch the runbook.
For example, if you want to launch the runbook every Monday, under **Days of the week** select **Monday** and under **Occurrence**, select **First**, **Second**, **Third**, **Fourth**, and **Last**.

▶ To configure the schedule for specific days in the month

- a. On the **Details** tab of the **New Schedule** dialog, select the **Days of month** radio button.
- b. In the **Days of the month** field, enter the date, or dates, you want to launch the runbook.

Separate multiple dates with a comma.

For example, if you want to launch the runbook on the first and 15th of every

month, enter **1, 15** in the **Days of the month** field.

▶ **To configure the schedule for specific hours**

- a. On the **Details** tab of the **New Schedule** dialog, select **Hours**.
- b. In the **Schedule Hours** dialog, select the hours you want to launch the runbook.

You can both allow and deny the launching of a runbook during any period. For example, if you want to launch a runbook only outside of business hours, select the hours of 9am to 5pm for Monday, Tuesday, Wednesday, Thursday, and Friday, and then click **Denied**.

5. On the **Exceptions** tab of the **New Schedule** dialog, add any date exceptions for the runbook.
6. Click **OK**.
7. Click **Finish**.



Important

The scheduled date and time to launch a runbook is based on the system clock of the Runbook server. This enables schedules to function in virtual machine environments and to continue running even when the system clock adjusts for Daylight Savings Time.

▶ **To associate a schedule with a runbook**

1. In the **Runbook Properties** dialog, on the **General** tab, click the ellipsis (...) to browse for a **Schedule**.
2. Select a schedule, and click **OK**.
3. Click **Finish**.

Runbook Servers

This tab displays the list of Runbook servers assigned to run this runbook. If the list is empty, the runbook uses the setting defined in the **Runbook Servers** folder in the **Connections** pane. If the Runbook server with the Primary role is available, then the runbook runs on it. If the Primary Runbook server is not available, then each Runbook server with a Standby role will be checked until one is located that can run the runbook.

You can override the default behavior and assign a Primary and any number of Standby Runbook servers to a runbook. It is useful to assign a specific Runbook server to a runbook if the runbook requires access to a specialized resource, such as a backup device.

▶ **To assign Primary and Standby Runbook servers to a runbook**

1. In the **Runbook Properties** dialog, on the **Runbook Servers** tab, select **Override**

- default Runbook Server roles** to configure primary and standby Runbook servers.
2. Click **Add**.
 3. Select a Runbook server, and then click **OK**.
The first Runbook server added becomes the Primary Runbook server.
 4. To add additional Runbook servers, click **Add**, and select another Runbook server.
All additional Runbook servers are added as Standby Runbook servers.
 5. When you are finished adding Runbook servers, click **Finish**.

Logging

This feature controls what data is logged to the Orchestrator database. If stored in the database, this data is visible in views such as the **Log** pane in the Runbook Designer and in the Orchestration console. This information has no effect on the availability of published data within a running runbook.

Published Data includes data specific to each activity. For details on the published data available in each of the standard activities, see the **System Center Orchestrator 2012 Runbook Activity Reference**. For details on published data available in integration packs, see the guides at **System Center Orchestrator 2012 Integration Packs**.

Common Published Data is a set of data items that are common to all activities. These items 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



Caution

Turning on logging increases the size of the Orchestrator database.

Event Notifications

You can enable event notification for the runbook. Notifications appear in views such as the Log pane in the Runbook Designer and in the Orchestration console.

If you want to be notified when a runbook runs for more than a specified length of time, enter a value in the **seconds** field.

If you want to be notified if the runbook fails to run, select the **runbook fails to run** option.

For more information on Event Notifications, see [Troubleshooting](#).

Job Concurrency

The job concurrency setting enables you to set the maximum number of simultaneous jobs, allowing you to execute multiple requests for the same runbook at the same time. This setting

applies to the individual runbook. A Runbook server can run 50 runbooks simultaneously. If you choose a job concurrency setting over 50, your environment requires more Runbook servers or the requests to launch a runbook will queue.

The following limitations apply:

- You cannot execute simultaneous requests for runbooks that start with Monitor activities. If you try to change the maximum number of simultaneous requests for these runbooks, the Runbook Designer resets the **Maximum number of simultaneous jobs** to 1 and displays an error message.
- Runbook server will execute simultaneous requests for runbooks up to the maximum processing limit. To change the maximum processing limit, see **How to Configure Runbook Throttling** in the System Center Orchestrator 2012 Deployment Guide.
- Do not create simultaneous requests for runbooks that contain Modify Counter activities. Simultaneous executions of the same runbook that modify (set, reset, increment, or decrement) a Counter can cause the Counter value to become unreliable. You can read the value of Counters in runbooks that execute simultaneously.
- Do not run simultaneous requests for runbooks that interact with a third-party product, such as a ticketing or system-monitoring tool, unless you have a thorough understanding of how the tool handles parallel processing. If the third-party application cannot handle parallel processing, or if you do not know, leave the maximum number of simultaneous requests at a value of 1.
- Plan the use of multiple requests carefully. Before you change the maximum number of simultaneous runbook requests, consider the tasks performed by the runbook. Verify that each runbook instance will be able to complete successfully. For example, if your runbook creates a folder, places files into it, and then deletes the folder when it is finished, one instance of the runbook may delete the folder before other instances are finished with it. In this case, you should keep the maximum number of simultaneous requests for this runbook a value of 1 to avoid conflicts.

Returned Data

Returned Data defines the data a runbook returns when it completes. Each returned data definition can contain either a single or multiple parameter values. To populate the data definitions, end the workflow with a Return Data activity that contains the return values.

You access the Returned Data values through Published Data in one of several ways.

- Invoke the runbook from within another runbook using the Invoke Runbook activity. The parent runbook can access the child runbook's Returned Data as Published Data from the Invoke Runbook activity.
- View the Published Data from the Runbook Designer or Orchestration console.
- Use the Orchestrator web service to return the Published Data programmatically.

For more information about the standard activities Invoke Runbook, Initialize Data, and Return Data, see the **System Center Orchestrator 2012 Runbook Activity Reference**.

To define the Returned Data for a runbook to return, use **Add**, **Edit**, and **Remove** to create each parameter.

Runbook Permissions

Runbook access permissions are set through the Runbook Designer. By default, only users in the Orchestrator Users Group have full access to a runbook. You give access to additional users to run, start, stop, view, and change runbooks at either the folder level or the individual runbook level.

▶ To view or modify the permissions of a runbook

1. In the Runbook Designer, in the **Connections** pane, click the **Runbooks** folder.
2. In the Runbook Designer workspace, right-click the tab of a runbook, and then select **Permissions**.
3. To give another user or security group access to the runbook, click the **Add** button and select the user or security group from the local computer or from the domain.
4. If the user or security group should be able to view and run the runbook, select the **Allow** box next to **Read**.
If the user or security group should be able to change the runbook, select the **Allow** box next to **Write**.
If the user or security group should be able to change permissions for the runbook, select the **Allow** box next to **Full Control**.
5. To close the **Permissions for Runbook** dialog and save any changes, click **OK**.

Activities

In System Center Orchestrator 2012, activities are the building blocks of runbooks. In general, individual activities perform three actions:

1. Access Published Data.
2. Perform some action.
3. Publish new data.

All activities, regardless of origin or type, share common behaviors. This section describes the types of activities available in Orchestrator and their common behaviors.

Activities

- [Standard Activities](#)
Standard activities available in System Center Orchestrator 2012.
- [Monitors](#)
Specialized activities that monitor environment states and event logs.
- [Customized Activities](#)
Customization options available in Orchestrator.
- [Common Activity Properties](#)

Configurable properties common to all activities.

Other resources for this product

- TechNet Library main page for **System Center Orchestrator 2012**
- [System Center Orchestrator 2012 Runbook Guide](#)
- [Runbook Concepts](#)
- [Runbooks](#)
- [Workflow Control](#)

Standard Activities

An installation of System Center Orchestrator 2012 includes a set of standard activities. Using these activities, you can create very powerful workflows to automate tasks and processes.

For detailed reference information on each of the standard activities, see **Standard Activities** in the System Center Orchestrator 2012 Runbook Activity Reference.

Standard activities are organized into categories to help you locate the appropriate activity for the task that you want to perform. The categories for standard activities are:

Category	Description
Email	Supports sending email notifications.
File Management	Manages file interaction such as copying and moving files.
Monitoring Activities	Reacts to system-level events.
Notification	Supports other notification types such as Syslog files and pop-up notifications.
Runbook Control	Manages how runbook logic behaves.
Scheduling Activities	Performs schedule-based activities.
System Activities	Runs system commands such as running a program.
Text File Management	Manipulates text files.
Utilities	Supports working and manipulating data within a workflow.

Monitors

Monitors are a specialized type of activity that is triggered by a state or event of an object outside of a runbook. For example, a monitor can wait for a particular event to occur in an event log, check the IP status of a certain computer, or simply run repeatedly on a predefined schedule.

An Orchestrator activity cannot trigger a monitor. Monitors are a start condition within a runbook. The monitor activity waits for the event trigger. When an event occurs, the monitor triggers the next activity in the runbook workflow.

Runbooks that start with a monitor activity load the monitor activity and wait for the trigger condition to occur. Once the monitor detects the trigger condition, a runbook instance is created to run the remaining activities. The monitor continues to run and waits for the trigger event to reoccur. Runbooks that start with monitors continue to run until you stop them from the **Runbook Designer** or **Orchestration console**.

For a list of standard monitor activities, see **Monitoring Activities** in the System Center Orchestrator 2012 Runbook Activity Reference.

Customized Activities

System Center Orchestrator 2012 provides two options for extending standard activities. Integration packs (IP) are Microsoft and third-party products that contain additional activities specific to a product or technology. For more information about the currently available IPs, see the TechNet topic **System Center Orchestrator 2012 Integration Packs**. If the functionality you require is not available in an IP, you have the alternative option of using the Orchestrator Integration Toolkit. This toolkit allows you to build an activity to meet your requirements. For more information about the Orchestrator Integration Toolkit, see the MSDN topic [System Center Orchestrator 2012 SDK](#).

Common Activity Properties

All activities have properties. The properties dialog box for each activity has multiple tabs that provide access to the settings for the activity. The particular set of tabs varies between activities, but there are several common property types.

Details Tab

This tab contains various properties specific to an activity. Many activities require you to at least enter a computer name, IP address, file name, file path, or file folder location. Details on these options are provided in the reference for each activity in **Standard Activities**.

Run Behavior

This tab contains the properties that determine how the activity handles multi-value Published Data. It also defines the notifications created if the activity fails or runs for an excessive period.

Returned Data Behavior

By default, Published Data is passed as multiple individual outputs. You can alternatively specify that the all values be flattened into a single comma delimited value.

When you enable the **Flatten** feature, you also choose a multi-value formatting option.



Note

The **Flatten** feature does not flatten data across multiple instances of the same activity. It only flattens multiple values returned from a single instance of the activity.

Flatten Behavior	Description
Separate with line breaks	Each item is on a separate line. This is the format for the output text files.
Separate with	Each item is separated by one or more characters, for example, a semicolon (;).
Use CSV format	All items are in CSV (comma-separated value) format, which is useful for importing into spreadsheets or databases.

Event Notifications

Some activities are expected to take a limited amount of time to complete. If the activity does not complete within the specified period, the activity can be stalled or another issue prevents the activity from completing. You can define the number of seconds to wait for completion of the activity, after which a platform event is sent to report the delay in completion. You can also choose whether to generate a platform event if the activity returns a failure. For more information on event notifications, see [Troubleshooting](#).

Event Notification Setting	Description
Report when the activity runs for more than:	In the edit field provided, enter the number of seconds of run time to elapse before generating a notification.
Report if the activity fails to run	Select this option to generate run failure notifications.

Security Credentials Tab

The settings on the **Security Credentials** tab enable you to specify w the account that runs the runbook server service. This is useful when the activity performs activities with resources on a remote computer.

Use the security of the account assigned to the service	Select this option to run the activity with the account used by the Runbook server. For more information, see Orchestrator Security

	Planning.
This account	Select this option to run this activity with another account. Specify the account user name and password to run this activity. Verify that the account has the rights to perform this action. If the credentials you provided fail validation, the account assigned to the Runbook server account is used.

Workflow Control

When you build runbooks in System Center Orchestrator 2012, it is important to understand the underlying logic of the workflow engine. By using this logic, you can create workflows to automate resource-based jobs and complex data processing tasks.

Workflow Control

- [Starting Point](#)
- [Smart Links](#)
- [Embedded Loops](#)

Other resources for this product

- TechNet Library main page for **System Center Orchestrator 2012**
- [System Center Orchestrator 2012 Runbook Guide](#)
- [Runbook Concepts](#)
- [Runbooks](#)
- [Activities](#)

Starting Point

A runbook can only have one starting point. A starting point is an activity that automatically runs when the runbook is started. Each activity in the runbook runs after the previous activity in the workflow is complete.

If a runbook starts with any activity other than a monitor, the runbook begins processing and attempts to run to completion. If the runbook starts with a monitor activity, the monitor loads and waits for the trigger condition. Once the condition is met, a runbook instance is created to run the remaining activities in the runbook. The monitor continues to run and wait another occurrence of the trigger condition. Runbooks that start with monitors continue to run until you stop them from the Runbook Designer or Orchestration console.

See Also

[Monitors](#)

Smart Links

The links that connect individual activities in a runbook are called smart links. Smart links in System Center Orchestrator 2012 allow precedence between two activities. Smart links invoke the next activity in the runbook as soon as the previous activity completes successfully. Smart links also provide filtering capabilities for the data so you can limit the data passed to subsequent activities in the workflow.

Creating and configuring smart links

You can modify the smart link condition properties by double-clicking the smart link.

Use the following procedure to enable or disable smart links.

▶ To create a smart link

1. In the Runbook Designer workspace, click and drag two activities from the **Activities** pane to the Runbook Designer workspace.
2. In the Runbook Designer workspace, hover the mouse cursor over one of the activities, click the right arrow, and then drag it to the destination activity.

A line is created between the two activities indicating a smart link is created.

▶ To disable a smart link connection while preserving configured properties

1. Right-click the smart link, and then select **Enable** to disable the smart link.
The smart link changes to a dashed line indicating that it is disabled.
- 2.

▶ To enable a smart link connection

1. Right-click a disabled smart link, and then select **Enable** to enable the smart link.
The smart link changes to a solid line to indicate that it is enabled.

General Tab

From the General tab in the Link Properties dialog box, you can add Name and Description values to the smart link. These properties are not required, but are useful in identifying the purpose of the smart link. These properties are not displayed unless you configure the runbook option to show link labels.

▶ To add a smart link label from the Runbook Designer

1. To view the smart link name, right-click a smart link, and select **Properties** from the menu.
2. In the **Properties** dialog, on the **General** tab, in the **Name** box, enter a descriptive name.

3. Click **Finish**.

▶ **To display smart link names in the runbook**

1. From the **Runbook Designer** menu, select the **Options** menu and select **Configure** to open the **Configuration** dialog box.
2. Select **Show link labels**.
3. Click **Finish**.

Include and Exclude Tabs

Orchestrator enables you to configure conditions for passing data to the following tasks in the runbook. Link conditions enable you to build branching capabilities into your runbooks. For example, a runbook must stop a database server before backing it up. If the database server stops correctly, the runbook starts the backup application. If the database does not stop correctly, an email is sent to the administrator to escalate the issue.

The **Include** tab allows you to specify the conditions that enable data to flow to the next activity in the runbook. The **Exclude** tab specifies the conditions that cause data to be excluded from the next activity in the runbook.

See [Smart Link Criteria](#) for details on using criteria in smart links.

 **Important**

The rules of the smart link **Exclude** tab supersede the rules on the smart link **Include** tab.

 **Important**

The rules on each tab are joined by using an **or** condition. Only one of the conditions defined on a tab must be true for the condition to be true.

Use the following procedure to add or remove a condition to a smart link.

▶ **To add a smart link condition**

1. Right-click a smart link, select **Properties** from the menu to open the **Link Properties** dialog box.

 **Important**

To change the values that make up the rule, you need to select each underlined portion of the smart link condition.

2. Click the listed activity in the condition to open the **Published Data** dialog box.
3. Select the **Show common Returned Data** box to display properties that are common to all activities.
4. Select a property from the published data and click **OK**. The criteria expression is changed depending on the type of data that the property returns.
5. To change the different parts of the expression, select the underlined text and either select or type in an appropriate value. For more information on criteria, see [Smart link](#)

[criteria](#).

6. Click **Finish**.

▶ To remove a smart link condition

1. From the **Link Properties** dialog box, select either the **Include** tab or **Exclude** tab.
2. To highlight the condition you want to remove, click to the right of the link condition on the word **or**, and then click **Remove**.
3. Click **Finish**.

Options Tab

From the **Options** tab in the **Link Properties** dialog box, you can specify different link colors on your branches to make them easier to read. For example, you can select green for the **Pass** branch and red for the **Fail** branch to identify the difference logic paths.

On this tab, you can also specify a delay before the activity runs.

Use the following procedure to configure these settings.

▶ To configure smart link colors

1. Click **Color**, and then click the color of the smart link.
2. Click **Width** of the smart link line in pixels.
3. Click **Finish**.

▶ To configure smart link activity delay

1. In the **Trigger delay** box, type the number of seconds that you want the smart link to wait before invoking the next step in the runbook.
2. Click **Finish**.

Smart link criteria

Link criteria can be created for any data published from the activity that initiates the link. The type of criteria will depend on the type of data returned from the particular property. The following sections provide details on the different types of data that activities can return.

Activity Completion Status

When you add a new criteria to the link, it will default to the completion status of the activity. This will return one of the following values:

- success
- warning
- failed

Each time you create a new link, it will create a default criteria specifying that the activity's completion status must return **success**. If you want the next activity to run regardless of whether the first activity successfully completes, then you should delete or change criteria.

Binary Values

Some properties will return a value of **true** or **false**. You can set a criteria of equals or does not equal, and the value will prompt you for the two possible values.

Text Values

Certain published data properties will return text that you can compare to an expected value or pattern. The different criteria that can be used are shown in the following table:

Condition	Description
contains	The specified text appears somewhere in the value of the published data item.
does not contain	The specified text does not appear somewhere in the value of the published data item.
starts with	The value of the published data item starts with the specified text.
ends with	The value of the published data item ends with the specified text.
matches pattern	The value of the published data item matches the specific regular expression.
does not match pattern	The value of the published data item matches the specific regular expression.
equals	The value of the published data item exactly matches the specified text.
does not equal	The value of the published data item does not match the specified text.



Note

Text values are not case sensitive.



Important

The regular expression criteria have a slightly different behavior than other regular expressions when using the ^ character specifying the starting position in the text and the \$ character specifying the ending position in the text. You must specify a wildcard in addition to these operators. For example, with the string “This is some sample text”, **text\$** will return a false, but **.*text\$** will return true. Similarly, **^This** will return false, but **^This.*** will return **true**.

Numeric Values

Certain published data properties will return numeric data that you can compare to an expected value. The different criteria that can be used are shown in the following table:

Condition	Description
equals	The value of the published data item is exactly equal to the specified value.
does not equal	The value of the published data item does not equal the specified value.
is less than	The value of the published data item is less than the specified value.
is greater than	The value of the published data item is greater than the specified value.
is less than or equal to	The value of the published data item is less than or equal to the specified value.
is greater than or equal to	The value of the published data item is greater than or equal to the specified value.
is between	The value of the published data item is between two specified values.

Embedded Loops

In System Center Orchestrator 2012, looping can be configured for any runbook. By using loops, you can build automatic retries and monitor at any location in a runbook.

Each activity can create a loop so that you can retry operations if they fail or test the output information of the activity for valid data. You can also use these mechanisms to build wait conditions into your workflows.

When a loop is configured for an activity, it continues to run with the same input data until a desired exit looping criteria is reached. The exit criteria is built in a similar way to smart link configurations. You can use any published data item from the activity as part of the exit or do not exit configuration. Included in the common published data are special data items such as **Loop: Number of attempts** and **Loop: Total duration** which allow you to use information from the loop itself in the looping conditions.

Loops run once for each incoming piece of data passed to the activity. For example, consider a runbook that uses a **Query Database** activity followed by **Append Line**. If the **Query Database** activity returned three rows, the **Append Line** activity would run three times. If you have a loop on the **Append Line** activity, it would run three separate loops. After the first data item has looped through the **Append Line** activity, then the next item goes through **Append Line** and loops until it exits, and then the third. After all three items have been processed, then the next activity in the runbook runs.

Configuring Looping

Use the following procedure to configure looping.

▶ To configure looping

1. Right-click an activity in the runbook, and then click **Looping** to open the **Looping Properties** dialog box.
2. On the **General** tab click **Enable**.
3. In the **Delay between attempts** box, type the number of seconds to pause between each attempt to run the activity.

Exit and Do Not Exit Tabs

The **Exit** tab specifies the conditions that determine whether the loop exits. The **Do Not Exit** tab specifies the conditions that cause the loop to continue.

Important

The rules on the **Do Not Exit** tab supersede the rules on the **Exit** tab.

The rules within each tab are joined by using an **Or** condition. Only one of the conditions on a tab must be true for the entire tab to be true.

Use the following procedure to add or remove an **Exit** condition.

▶ To add an exit condition

1. From the **Looping Properties** dialog box, select either the **Exit** tab or **Do Not Exit** tab, and then select the condition listed in the box or click **Add** to add a condition.

Important

To change the values that make up the rule, you need to select each underlined portion of the link condition.

2. Click the listed activity in the condition to open the **Published Data** dialog box.
3. Check the **Show common Returned Data** box to display properties that are common to all activities.
4. Select a property from the published data and click **OK**. The criteria expression is changed depending on the type of data that the property returns.
5. To change the different parts of the expression, select the underlined text and either select or type in an appropriate value. For more information on criteria, see [Smart Link Criteria](#).
6. Click **Finish**.

▶ To remove an exit condition

1. From the **Looping Properties** dialog box, select either the **Exit** tab or **Do Not Exit** tab.
2. To highlight the condition you want to remove, click to the right of the link condition with the word **Or**, and then click **Remove**.

3. Click **Finish**.

See Also

[Smart Link Criteria](#)

Tools

To create and test a runbook in System Center Orchestrator 2012 you use the Runbook Designer and the Runbook Tester.

- [Runbook Designer](#)
- [Runbook Tester](#)

Other resources for this product

- TechNet Library main page for **System Center Orchestrator 2012**
- [System Center Orchestrator 2012 Runbook Guide](#)
- [Runbook Concepts](#)
- [Design and Build Runbooks](#)
- [Deploy and Launch Runbooks](#)

Runbook Designer

The **Runbook Designer** is the tool that you use to create, manage, and run runbooks in System Center Orchestrator 2012. The **Runbook Designer** is intended for users who must create or modify runbooks. Users who only need to run runbooks and view their status should use the **Orchestration Console** which is documented in the **System Center Orchestrator 2012 Orchestration Console Guide**.

Runbook Designer Panes

The Runbook Designer interface is organized into the following four panes.

Pane	Description
Connections	The folder structure where you can organize workflows in the Orchestrator system and edit permissions on folders. Also provides access to Runbook Servers and Global Settings .
Design Workspace	Workspace where you build Orchestrator runbooks. The runbooks in the folder selection in the Connections pane will be listed as tabs across the top of the workspace. When you select a runbook's tab, it will be displayed in the

Pane	Description
	Design Workspace.
Activity pane	Contains all the activities available (either standard activities or activities available from integration packs) for use in runbooks. You drag activities from the Activity pane into the Designing Workspace, then link them together to form runbooks.
Log	Logs showing the activity and history for the current runbook. For more information, see Troubleshooting .

Sorting Activities by Activity Name and Category Name

System Center Orchestrator 2012 enables you to sort activities alphabetically by activity name, or by category name. By default, activities are sorted by category, such as System, Scheduling, Monitoring, File Management, Email, Notification, Utilities, Text File Management, and Runbook Control.

Use the following steps to sort activities by their activity name and category name.

▶ To sort activities alphabetically by activity name

1. In the **Activity pane**, right-click a category name.
2. From the menu, select **All Activities**. The activities are sorted alphabetically by activity name.

▶ To sort activities by category name

1. In the **Activity pane**, right-click a category name.
2. From the menu, select **Default**. The activities are sorted alphabetically by category name.

Changing Icons

You can change the default size of each activity icon from small to large.

Use the following steps to change the icon size.

▶ To change the icon size

1. In the **Activity pane**, right-click an activity name.
2. From the menu, select **Small** or **Large** depending on the size of icon that you want to view.

Runbook Tester


Runbook Tester lets you test runbooks in a debugging environment. You can run an entire runbook, step through it one activity at a time, or add breakpoints to stop the simulation at any activity you select. You launch Runbook Tester from the button above the Design Workspace in the **Runbook Designer**. When you click this button, the Runbook Tester will be started and load the current runbook. The button is only enabled if the runbook is not currently running. You must stop the runbook before you can test it.

Important

Runbook Tester actually performs each activity within the workflow. The steps are not performed in a simulated or virtualized environment. All the connections referenced in the runbook are live and fully functional, so any activities that modify or destroy data in connected systems cause that data to be modified or destroyed. For example, if you use the **Query Database** activity to **DROP TABLE ImportantTable**, it actually deletes the **ImportantTable** from the instance of SQL Server.

Runbook Tester panes

The Runbook Tester interface is organized into the following four panes.

Pane	Description
Run Time Properties	Displays run-time information, including resolved published data items, variables, and computer groups, about the activity that is currently being processed by Runbook Tester. Information appears in this pane when the runbook runs with breakpoints or in step-through mode.
Design Time Properties	Displays design-time information about each activity in the runbook when the runbook runs without breakpoints and is not run in step-through mode. To view the design-time properties of an activity, click an activity in the runbook.  Note You cannot edit the information that appears in the Design Time Properties pane.
Workspace	Displays the active runbook. You can select

Pane	Description
	each activity to view its information in the Design Time Properties pane or to set a breakpoint on it.
Log	Displays information about each activity in the runbook as it runs. You can click the Show Details link to show the configuration details and published data from the activity.
Resource Browser	Displays the counters, variables, computers groups, and schedules that the runbook in the workspace uses.

Design and Build Runbooks

This section provides details on how to design, build, test, and deploy runbooks with System Center Orchestrator 2012.

Runbooks

- [Designing a Runbook](#)
Design guidance for building runbooks.
- [Building a Runbook](#)
Create a runbook.
- [How to Test a Runbook](#)
Test the runbook.
- [Troubleshooting](#)
Tips and guidance on troubleshooting a runbook.

Other resources for this product

- TechNet Library main page for **System Center Orchestrator 2012**
- [System Center Orchestrator 2012 Runbook Guide](#)
- [Runbook Concepts](#)
- [Tools](#)
- [Deploy and Launch Runbooks](#)

Designing a Runbook

When you plan a new runbook, you should start with a defined process that you want to automate. This process determines your choice of runbook activities. Specifically, determine the following:

- When and how often is the runbook going to run?
- What steps make up the workflow?
- What activities reflect the steps in my workflow?.
- What type of data is required to begin the workflow?
- What data are generated from each activity?
- What results are produced at the end of the workflow?
- How are the runbook results reported?

Consider the following points as you design your runbook:

1. Failure and Warning links - It is important to handle all results from an activity. An activity provides a default success string, but does not provide a default failure case. Consider if you should reverse an activity or write the result to a log file.
2. Replace the default strings - When you look at the workflow in a runbook, the labels should identify what the individual activities are doing. Rename links and activities labels to a descriptive name.
3. Link colors - Change the color of your links when there is a condition or branch. It is common to use GREEN as success and RED for warning or failed. You should use standard associations, but not use too many colors or you lose the descriptive purpose.
4. Limit the number of activities per runbook - Too many activities in a single runbook make it difficult to administer and troubleshoot. Consider splitting a runbook into several subtasks and create child runbooks for each of those subtasks. You can invoke the child runbooks from a parent runbook. You can reuse these child runbooks in other workflows.
5. Runbook logs - By default, logging options are disabled for runbooks. When you enable logging, the data significantly increases the size of your database. As an alternative, you can log to an external system or file.

Building a Runbook

This topic describes the basic process for creating an System Center Orchestrator 2012 runbook.



Note

For a list of topics that contain more details about the information covered here, see [Runbook data processing](#).

Step	Description
1. Create runbook	Create an empty runbook in the Runbook Designer.
2. Add activities	Click and drag activities from the Activities

Step	Description
	pane into the runbook. Include a start point and an end point for the runbook.
3. Link activities	Create and configure smart links between each of the activities to create a complete workflow.
4. Configure runbook properties	Configure the properties for the runbook.
5. Check in your runbook	Save your changes and check in the runbook.

Creating a runbook

▶ To create a new runbook

1. On the computer where the Runbook Designer is installed, click **Start**, click **All Programs**, click **System Center Orchestrator 2012**, and then click **Runbook Designer**.
2. In the Runbook Designer **Connections** pane, click the **Runbooks** folder.
3. In the **Connections** pane, click the **Create a new runbook** icon .
4. In the Runbook Designer workspace, right-click the runbook tab, and then select **Rename**.
5. In the **Confirm Check out** dialog box, click **Yes**.
6. Enter a name for the runbook, such as **Sample Runbook**, and press **Enter**.

▶ To add and configure activities to your runbook

1. In the **Activities** pane, click and drag an activity to the Designing pane of your runbook.
2. In the **Activities** pane, double click an activity to open the **Properties** dialog box for that activity.



Note

For information about specific properties of standard activities, see the **System Center Orchestrator 2012 Runbook Activity Reference**.

▶ To add and configure links in a runbook

1. To create a link, click and drag the arrow of an activity to another activity.
2. On the newly created link, double click the link to open the **Link Properties** dialog box.



Note

For information about the properties of links, see the **System Center Orchestrator 2012 Runbook Activity Reference**.

▶ To define the properties of a runbook

1. Right-click the **Runbook** tab and then click **Properties** to open the **Runbook Properties** dialog box.
2. Configure the settings on the **General** tab. The following tables provide the configuration instructions.
3. Click **Finish** to save your settings.

▶ **To check in your runbook**

1. In the Runbook Designer, click the **Check-in** icon on the toolbar.

Runbook data processing

- [Data Manipulation](#)
- [Published Data](#)

Other resources for this product

- TechNet Library main page for **System Center Orchestrator 2012**
- [System Center Orchestrator 2012 Runbook Guide](#)
- [Designing a Runbook](#)
- [How to Test a Runbook](#)
- [Troubleshooting](#)

Data Manipulation

With System Center Orchestrator 2012, you can manipulate string data from text files, returned data, or other sources, and convert it into a usable form. You can also perform simple arithmetic operations, such as calculating sums and differences, and performing division and multiplication operations. For example, you can extract text from a text file by using a text file management activity, trim leading and trailing spaces from the text, and then retrieve specific parts of the text that you can pass to other activities as returned data items.

Data Manipulation

- [Computer Groups](#)
- [Counters](#)
- [Functions](#)
- [Regular Expressions](#)
- [Schedules](#)
- [Variables](#)

Other resources for this product

- TechNet Library main page for **System Center Orchestrator 2012**
- [System Center Orchestrator 2012 Runbook Guide](#)
- [Building a Runbook](#)

- [Published Data](#)

Computer Groups

System Center Orchestrator 2012 is designed to interact with all of your data center systems. Computer groups enable you to target selected activities against a set of similar computer systems instead of a single computer. By configuring the activities in your runbook to use a computer group, you have the flexibility to add computers dynamically by adding them to the computer group.

Computer groups can be created with Active Directory queries. By using Active Directory queries, you can manage the list of computers in a group outside of Orchestrator. For example, if you have a computer group that is created from an Active Directory query that retrieves all the instances of Microsoft SQL Server, when an instance of SQL Server is added to your Active Directory system, it will automatically be included in that group.

Managing Computer Groups

To use computer groups in your activities, create a computer group, and then add computers to it. You can add computers by using the name of the computer, an Active Directory query, or a System Center Configuration Manager collection.

You can also organize your computer groups into folders. Use the following steps to create a new folder.

▶ To create a new folder

1. From the **Connections** pane in the Runbook Designer, click the **Computer Groups** folder or a sub-folder.
2. Right-click and from the menu select **New** and then **Folder**.

Use the following procedure to add a computer group.

▶ To add a computer group

1. From the **Connections** pane, right-click the **Computer Groups** folder or a subfolder.
2. Click **New**, and then click **Computer Group** to open the **New Computer Group** dialog box.
3. In the **New Computer Group** dialog box, on the **General** tab, type a **Name** and **Description** of the computer group.
4. Click the **Contents** tab. The list displays all the computer entries that make up this computer group.
5. Click **Add** to open the **Entry Type** dialog box.
6. Select the type of entry that you want to add to the computer group. You have the choice to select **Computer**, **Active Directory Query**, or **SMS Collection**.
 - If you selected **Computer**, type the name of the computer that you are adding, or click the ellipsis (...) button next to the **Computer** box, and then select the applicable computer. Click **OK** to add the computer. Click **Add** and repeat this step to add more computers to the group.

- If you selected **Active Directory Query**, click **OK** to open the **Enter a name** dialog box. Type the name of the entry that will be listed into the box, and then click **OK**.
In the **Find Computers** dialog box that opens create the Active Directory query. Click the **Find Now** button to see what computers match your query. Click **OK** to add the query.
- If you select **SMS Collection**, type the name of the server that you are adding, or click the ellipsis (...) button next to the **Server** box, and then select the applicable computer.

▶ To modify settings

1. To modify the settings of an entry you added, click the entry on the **Contents** tab, and then click **Modify**.
2. To remove an entry on the **Contents** tab, click the entry, and then click **Remove**.

Using a Computer Group in an Activity

Any standard activity that requires you to identify a **Computer** name in the configuration properties dialog box, such as the Send Event Log Message activity, can use a computer group. Other activities can use the **Computer Group** where you define a remote system or computer. Use the following procedure to use a computer group.

▶ To use a computer group

1. Right-click the applicable activity from your runbook, select **Properties** from the menu, and then select the **Details** tab to open the activities properties dialog box.
2. In the **Computer** box, right-click to open a menu, select **Subscribe**, and then select **Computer Group** to open the **Select Computer Group** dialog box.
3. Select the computer group, and then click **OK**.

A placeholder {computer group name} is inserted next to the computer name in the **Computer** box.

When the activity runs, it runs on each computer in the group.

Counters

When building runbooks in System Center Orchestrator 2012, you might find that there are values that must be incremented, such as keeping track of the number of backup attempts that a runbook made. Counters enable you to modify and check the status of a number that you can use to keep track of important statistics. You create a counter from the Connections pane in the Runbook Designer and then work with it using the **Get Counter Value** and **Modify Counter** activities. Each of these activities presents the value of the counter as [Published Data](#) so that it can be used by other activities and links.

Warning

You cannot run multiple, simultaneous jobs for runbooks that contain Modify Counter activities because simultaneous jobs of the same runbook that modify (set, reset, increment, or decrement) a counter can cause the counter value to become unreliable. You can, however, read the value of counters in runbooks that run simultaneously.

Use the following procedures to create a counter and organize counters.

▶ To create a counter

1. From the **Connections** pane, double-click the **Global Settings** folder, right-click the **Counters** folder or a subfolder of the **Counters** folder, point to **New**, and then click **Counter** to open the **New Counter** dialog box.
2. Type a **Name** for the counter.
3. Type a **Description** that explains the purpose of the counter.
4. Type the **Default Value** of the counter. This value is the starting value of the counter when it is created or reset.
5. To modify a counter, double-click the counter.
To remove a counter, right-click the counter, and then select **Delete**.
6. Click **Finish**.

▶ To organize counters

1. You can group counters into folders to organize them. To create a folder, right-click the **Counters** folder, point to **New**, and then click **Folder**.
2. To move a counter to a different folder, right-click the counter, and then click **Move** to open the **Select a Folder** dialog box.
3. Select the destination folder, and then click **OK**. The counter is moved to the new folder location.

Functions

With System Center Orchestrator 2012, you can manipulate string data from text files, returned data, or other sources, and convert it into a usable form. You can also perform simple arithmetic operations, such as calculating sums and differences, and performing division and multiplication operations. For example, you can extract text from a text file by using a text file management activity, trim leading and trailing spaces from the text, and then retrieve specific parts of the text that you can pass to other activities as returned data items.

For a complete list of the functions that you can perform, see the following Functions table.

Data Manipulation Functions

You can insert a data manipulation function into any box that allows you to type text. Data manipulation functions must be enclosed in square brackets ('[' and ']'). For example:

```
[Upper('this will be inserted in upper case')]
```

When the activity runs, the text 'this will be inserted in uppercase' in the example is replaced with 'THIS WILL BE INSERTED IN UPPERCASE'.

Nested Functions

If you want to use a data manipulation function within another function, you do not have to enclose the nested function in square brackets. For example, to nest the **Field** function, type:

```
[Field(Field('username=jsmith@abcompany.com', '=', 2), '@', 1)]
```

Functions

Functions are case-sensitive. For example, Upper('Text') will be processed, but upper('Text') will not.

Function and Definition	Usage	Parameters	Example
Upper - converts text to uppercase.	Upper('Text')	Text - the text that is being converted to uppercase.	Upper('this will be converted to uppercase') returns 'THIS WILL BE CONVERTED TO UPPERCASE'
Lower - converts text to lowercase	Lower('Text')	Text - the text that is being converted to lowercase.	Lower('This Will Be Converted To Lowercase') returns 'this will be converted to lowercase'
Field - returns text in a specific position	Field('Text', 'Delimiter', Field Number)	Text - the text that is being searched Delimiter - the character that separates each field Field Number - the position of the field that is being returned (starting at 1)	Field('John;Smith;9055552211', ';', 2) returns 'Smith'.
Sum - returns the sum of a set of numbers	Sum(firstNumber, secondNumber, thirdNumber, ...)	Number - the number that is being added, you can put any set of numbers, each separated by a	Sum(2,3,4,5) returns '14'

Function and Definition	Usage	Parameters	Example
		comma (,).	
Diff - returns the difference of two numbers	Diff(Number1, Number2, <Precision>)	Number1 - the number that will be subtracted from Number2 - the number that will be subtracted from Number1 Precision <Optional> - the number of decimal places that the result will be rounded to	Diff(9, 7) returns '2' Diff(9.3, 2.1, 2) returns '7.20'
Mult - returns the product of a set of numbers	Mult(firstNumber, secondNumber, thirdNumber, ...)	Number - the number being multiplied, you can put any set of numbers each separated by a comma (,).	Mult(2, 3, 4) returns '24'
Div - returns the quotient of two numbers	Div(Number1, Number2, <Precision>)	Number1 - the number that will be divided Number2 - the number that will divide Number1 Precision <Optional> - the number of decimal places that the result will be rounded to	Div(8, 4) returns '2' Div(9, 2, 2) returns '4.50'
Instr - returns the position of first occurrence of text within	Instr ('SearchText', 'TextToFind')	SearchText - the text that is being searched TextToFind - the	Instr('This is a string that is searched', 'string') returns 11

Function and Definition	Usage	Parameters	Example
another text		text that you are searching for	
Right - returns a subset of the text from the right side of the full text	Right('Text', Length)	Text - the full text Length - the number of characters from the right side that will be returned	Right('Take from the right', 9) returns 'the right'
Left - returns a subset of the text from the left side of the full text	Left('Text', Length)	Text - the full text Length - the number of characters from the left side that will be returned	Left('Take from the left', 4) returns 'Take'
Mid - returns a subset of the text from the middle of the full text	Mid('Text', Start, Length)	Text - the full text Start - the starting position in the text where you want to begin returning characters Length - the number of characters starting from the Start position that will be returned	Mid('Take from the middle', 5, 4) returns 'from'
LTrim - Trims leading spaces from text	LTrim('Text')	Text - the text being trimmed of leading spaces	LTrim(' Remove the leading spaces only. ') returns 'Remove the leading spaces only. '
RTrim - Trims the trailing spaces from text	RTrim('Text')	Text - the text being trimmed of trailing spaces	RTrim(' Remove the trailing spaces only. ') returns ' Remove the trailing spaces only.'
Trim - Trims leading and trailing spaces	Trim('Text')	Text - the text being trimmed	Trim(' Remove leading and trailing spaces. ') returns 'Remove leading and trailing spaces.'

Function and Definition	Usage	Parameters	Example
from text			
Len - returns the length of text	Len('Text')	Text - the text that is being measured	Len('Measure this text') returns 17

Regular Expressions

In System Center Orchestrator 2012, regular expressions enable you to match a string to a pattern. The regular expression can contain a number of different elements that define the pattern. **Smart Link Properties** use regular expressions to perform pattern matching.

Advanced Regular Expressions

To build regular expressions, you must create an expression that contains the text that you are searching for and special characters that create a pattern which describes how the text that you are searching for appears.

Character	Meaning
.	Matches any character except a newline.
*	Matches the preceding item 0 or more times. For example, the "a*" pattern matches any string of a's in a row "a", "aaa", "aaaaaaaaaaaa", and an empty string "". To match any string of any character, use a dot followed by an asterisk. For example "a.*" matches any text that begins with the letter "a" and ends with any string of characters such as "abbb", "abcdef", or "automatic restart".
+	Matches the preceding item 1 or more times. This is like * but you must have a least 1 of the preceding item to make a match. For example, the "ab+" pattern matches "abbbb", "ab", but does not match "a". To contrast, the "ab*" pattern matches "a".
?	Matches the preceding item 0 or 1 time. For example, the "ab?" pattern matches "a" or "ab" but does not match "abbb".
	Matches either the preceding expression or the following expression. Logical OR operator.

Character	Meaning
\$	Matches the expression at the end of the input or line. For example, "ab\$" matches "I took a cab" or "drab" but does not match "absolutely not".
^	Matches the expression at the beginning of the input or line. For example, "^ab" matches "absolutely not" or "abacuses are great!" but does not match "I took a cab" or "drab".
\	For characters that are usually treated as special. This indicates that the next character is literal and is not to be treated as a special character. For example, "\." means match the "." character and not just any character.
[]	A character set. Matches any one of the enclosed characters. You can specify a range of characters by using a hyphen. For example, [a-zA-Z] matches any letter of the alphabet.
[^]	An excluded character set. This is the opposite of []. If any of the characters inside the brackets exist, the regular expression match fails. You can specify a range of characters by using a hyphen. For example, [^a-zA-Z] makes sure that none of the letters in the alphabet are present.
()	A group expression. This groups an expression into an item that you can apply special characters to. For example, "a*(ba)+" matches "ba" "aba" or "ababa" but does not match "abbba" or "abaa"

Examples

Expression	Meaning
[a-zA-Z]+	The text contains only letters of the alphabet.
^*	The text begins with an asterisk.
(abc def)\$	The end of the text is either "abc" or "def".

Expression	Meaning
Ha..y	The text begins with "Ha" followed by any two characters followed by a "y".
Help.*	The text is "Help" followed by any number of other characters.

Schedules

System Center Orchestrator 2012 uses schedules to define the times when runbooks can run. For example, there are times when it is inappropriate to run some runbooks, such as backing up a runbook on a main server during regular business hours. You can create a schedule that will run according to a complex interval, such as the first and third Mondays and Thursdays of every month, except when these days fall on a holiday.

Schedules use the system clock of the Runbook server that runs the runbook. This enables schedules to function in virtual machine environments, and to continue running even when the system clock is adjusted because of the move to or from Daylight Savings Time.

Runbooks that start before a prohibited time run until finished, even if they are still processing when the prohibited time arrives. They will not be interrupted after processing has started.

Important

If a runbook is scheduled to start during an hour that is skipped when the system clock is adjusted forward by one hour, that starting time is skipped, and the runbook starts at the next scheduled time. If a runbook is scheduled to start during an hour that occurs two times because the system clock is adjusted backward by one hour, the runbook starts two times.

Conditional Links

In addition to assigning a schedule to a runbook, you can use a **Check Schedule** activity to use a schedule for conditional logic in a runbook. This activity will check a particular schedule and return a published data item with a value of true or false specifying whether the current time is within the schedule. This published data item can be used by a link to determine whether to run a particular activity or to continue to the workflow.

Creating a Schedule and Assigning the Schedule to a Runbook

Use the following procedures to create a schedule, to assign the schedule to a runbook, or remove a schedule from a runbook.

To create a schedule

1. From the **Connections** pane, right-click the **Schedules** folder or a subfolder of the **Schedules** folder, point to **New**, and then click **Schedule** to open the **New Schedule** dialog box.
2. On the **General** tab, type a **Name** for the schedule.

3. Type a **Description** that describes or explains the purpose of the schedule.
4. Click the **Details** tab. Select the days that this schedule allows runbooks to run:

Days of week: Select this option and select the days of the week when this schedule allows runbooks to run.

Occurrence: Select the weeks of the month when the schedule allows runbooks to run.

Days of month: Select this option and select the days of the month when this schedule allows runbooks to run. Specify the days of the month by entering the number of the day. You can use hyphens to describe ranges and commas to separate entries. For example, typing **1,3** will include the first and third day of the month. Typing **1-21** will include the first through to the twenty-first day of the month. You can combine both to create complex descriptions of the days of the month. Type **all** to specify all days of the month. Type **last** to specify the last day of the month.

You cannot use **all** and **last** as part of a range of days. Additionally, if you typed a range of 5-31, this range works correctly for all months, including those with 28, 29, 30, and 31 days.
5. Click **Hours** to open the **Schedule Hours** dialog box.
6. Click and drag to select a group of hours in a week. The text at the bottom of the dialog box shows the time period that you selected. Then select one of the following:

Permit (blue): assigns the time period that you selected as a time when runbooks are allowed to run.

Denied (white): assign the time period that you selected as a time when runbooks are not allowed to run.
7. Click **OK**.
8. Click the **Exceptions** tab. The list displays all the days that are exceptions to the rules defined in the **Details** tab.
9. Click **Add** to open the **Date** dialog box.
10. Specify the date and select **Allow** or **Disallow** to allow or not allow the runbook to run on that day, and then click **OK**. The entry appears in the list.
11. To modify an Exception entry, select it, and then click **Modify**. To remove the Exception entry, select it, and then click **Remove**.
12. To modify a schedule, double-click the **Schedule**.
13. To remove a schedule, right-click the **Schedule**, and then select **Delete**.
14. Click **Finish**.

 **To assign a schedule to a runbook**

1. Right-click the Runbook tab, and then click **Properties** to open the **Runbook Properties** dialog box.
2. On the **General** tab, click the ellipsis (...) button to open the **Select a Schedule** dialog box.
3. Select the schedule that you want to apply to the runbook, and then click **OK**.

4. Click **Finish**.

Every time the runbook is started, it checks the schedule to verify that it is allowed to run. If it is not allowed to run, it stops and does not restart until the next time it is started.

▶ **To remove a schedule from a runbook**

1. Right-click the Runbook tab, and then click **Properties** to open the **Runbook Properties** dialog box.
2. On the **General** tab, click the ellipsis (...) button to open the **Select a Schedule** dialog box.
3. Do not select a schedule. Click **OK**.
4. Click **Finish**. The schedule is removed from the runbook.

See Also

[Published Data](#)

Variables

When building runbooks in System Center Orchestrator 2012, there are settings that are the same across activities. Variables enable you to specify a value that is used by activities in any runbook.

Use the following procedures to create, insert, and organize variables.

▶ **To create a variable**

1. From the Connections pane in the Runbook Designer, expand the Global Settings folder, and click the Variables folder.
2. Right-click the **Variables** folder or a subfolder of the **Variables** folder, point to **New**, and then click **Variable** to open the **New Variable** dialog box.
3. Type a **Name** for the variable.
4. Type a **Description** that explains the purpose of the variable.
5. Type the **Value** of the variable. This value replaces the placeholder in those activities where the variable is inserted.
6. Click **Finish**.

▶ **To insert a variable in an activity**

1. Right-click the applicable activity from your runbook, from the menu select **Properties**, and then select the **Details** tab to open the activities properties dialog box.
2. In a text box, right-click to open a menu, select **Subscribe**, and then select **Variable** to open the **Select a Variable** dialog box.
3. Select the variable name, and then click **OK**.

A placeholder {variable} is inserted next to the computer name in the **Computer** box.

When the activity runs, the placeholder is replaced with the value of the variable.

► To organize variables

1. You can group variables into folders to organize them. To create a folder, right-click the **Variables** folder, point to **New**, and then click **Folder**.
2. To move a variable to a different folder, right-click the variable, and then click **Move** to open the **Select a Folder** dialog box.
3. Select the destination folder, and then click **OK**. The variable is moved to the new folder location.

Special Variables

You can specify special formats of variables to provide dynamic information to your runbooks. Specify the value of the variable to invoke this behavior.

NOW(): When the variable is resolved, it is set to the current date and time. You can pass arguments to this function to return specific portions of the date or time. For example, **NOW(hour)** returns the current hour. The following are the valid arguments for the **NOW()** function: day, dayofweek, dayofyear, month, year, hour, minute, second, millisecond.

%ENVVAR%: Returns the value of the environment variable between the percent (%) symbols. The environment variable is based on the runbook server computer where the runbook is running, and it is not case-sensitive. All system variables can be resolved. Any user variables are resolved in the context of the service account on the runbook server. If the environment variable does not exist, the text specified within the variable is returned as-is (that is, if you type **%ENVVAR%** and no environment variable named **ENVVAR** exists, then the text **'%ENVVAR%'** is returned).

Published Data

Published Data allows an activity to use information another activity in the same runbook. Each activity has a specific set of Published Data items that it populates after it runs. Any other activity that runs after it in the workflow has access to this data. In addition to data specific to each activity, all activities publish a common set of data items providing information such as the start and stop time of the activity and its completion status. Link conditions also use Published Data to add filtering and decision-making capabilities to runbooks.

For example, the runbook might use a **Read Line** activity to get information from a text file. A **Send Email** activity later in the runbook needs to use the information to include in the text of its mail. The **Send Email** activity could use the **Line Text** Published Data item from the **Read Line** activity to include in its mail message.

Data Types

The following table shows categories of Published Data value types.

Published Data Value Type	Description
String value	Text, for example, an error message description.

Published Data Value Type	Description
Date value	Date and time information. For example, the date and time that a specific error occurred.
Number value	Numeric information. For example, the number of rows returned by a database query.
Boolean value	true or false. For example, command completed.

Published Data with multi-value types

When an activity in a runbook runs, it runs one time for each item of data that the previous activity produced. For example, the Query Database activity runs and retrieves three rows from the database. These three rows of data cause the next activity to run three times, one time for each row returned. This next activity does not have to subscribe to the data for this action to occur.

An activity can also retrieve information from an outside source. **Get** activities and **Monitor** activities demonstrate this behavior. Data output from an activity might be a list of computers for example. Data can be passed on as multiple individual outputs, which invoke the next activity as many times as there are items in the output.

You also have the option of passing on data as a single output. For information about how to configure Published Data with multiple values, see [Common Activity Properties](#).

Adding Published Data to activity configurations

When an activity has subscribed to Published Data, a placeholder is inserted where the value of the data is added. An activity can only subscribe to Published Data from a previous activity in the workflow.

Use the following procedures to add Published Data to an activity, to change the Published Data subscription, and to copy and paste Published Data items.

▶ To subscribe to Published Data of an earlier activity in the workflow

1. Right-click an activity from your runbook, select **Properties** from the menu, and then select the **Details** tab to open the activity's properties dialog box.
2. From a text box, right-click to open a menu, select **Subscribe**, and then select **Published Data** to open the **Published Data** dialog box.
3. From the **Activity** drop-down list, select the activity that returns the data that you want to subscribe to. By default, the dialog box only displays Published Data that is specific to that activity. To include Published Data that is common to all activities, click **Show common Published Data**.
4. Select the **Published Data** item that you want to use, and then click **OK**.

▶ To change the Published Data subscription

1. From the text box click the data placeholder to open the **Published Data** dialog box.
2. From the **Activity** drop-down list, click the activity that returns the data that you want to subscribe to. By default, the dialog box only displays Published Data that is specific to that activity. To include Published Data that is common to all activities, click **Show common Published Data**.
3. Click the **Published Data** item that you want to use, and then click **OK**. The Published Data placeholder changes to reflect the new activity and Published Data that you selected.

 **To copy and paste Published Data items**

1. Find a Published Data item that has already been inserted into a box in the **Properties** dialog box of an activity.
2. Select the Published Data item that you want to copy.
3. Use the keyboard shortcut CTRL+C, or right-click the selected item, and then click **Copy**.
4. Open the **Properties** dialog box or document to which you want to copy the Published Data item.
5. Place your cursor where you want the Published Data item to appear and use the keyboard shortcut, CTRL+V, or right-click the insertion point, and then click **Paste**. The Published Data item appears.

Transforming Published Data items

You might need to modify text from a Published Data activity before using it in another activity. For example, you might need to remove a portion of the text or replace it with another string. You can transform the existing Published Data content or variable items into new content according to rules that you specify, by using the **Map Published Data** activity.

Common Published Data

The following table describes the Published Data items common to all activities.

Name	Description
Activity ID	The unique identifier of the activity. For example, {4BD3F27A-8F1B-4F60-8245-F69469075EF1}.
Activity name	The name of the activity as it appears in the workspace. If you customize the name of an activity in the workspace, the customized name appears here.
Activity Process ID	The process ID of the job process where the activity runs.
Activity status	The result status of running the activity, for

Name	Description
	example, Success.
Activity type	The default name of the activity. It does not change from the default even if you rename the activity in the workspace, and it can be useful in identifying an activity in runbooks where activity names and display icons have been changed.
Error summary text	A summary of the error information returned by the activity.
Runbook name	
Runbook Process ID	<p>The process ID of the runbook module's executable program running on the runbook server.</p> <p>The job process contains the logic for the activity. It is started when the runbook server starts the runbook, and it is stopped when the runbook is stopped. Each runbook runs in its own job process executable program.</p>
Server name	The name of the runbook server where the runbook is running.
Activity duration	The total time that activity was running.
Activity end time	The time when the activity finished.
Activity end time(year)	The year when the activity finished.
Activity end time(month)	The month when the activity finished.
Activity end time(day)	The day when the activity finished.
Activity end time(weekday)	The day of the week when the activity finished.
Activity end time(hours)	The hour when the activity finished.
Activity end time(minutes)	The number of minutes past the hour when the activity finished.
Activity end time(seconds)	The number of seconds past the minute when the activity finished.
Activity end time in UTC	The time when the activity finished in UTC format.
Activity end time in UTC (year)	The year when the activity finished in UTC

Name	Description
	format.
Activity end time in UTC (month)	The month when the activity finished in UTC format.
Activity end time in UTC (day)	The day when the activity finished in UTC format.
Activity end time in UTC (weekday)	The day of the week when the activity finished in UTC format.
Activity end time in UTC (hours)	The hour when the activity finished in UTC format.
Activity end time in UTC (minutes)	The number of minutes past the hour when the activity finished in UTC format.
Activity end time in UTC (seconds)	The number of seconds past the minute when the activity finished in UTC format.
Activity start time	The time when the activity started.
Activity start time in UTC	The time when the activity started in UTC format.
Loop: Delay between attempts	The amount of time (in seconds) between each loop attempt.
Loop: Enabled	The setting that determines whether per-activity looping is enabled for the activity.
Loop: Loop error message	The error message if the loop is not successful.
Loop: Number of attempts	The number of iterations that the loop has been through.
	The name of the runbook to which that the activity belongs.
Loop: Total duration	The total amount of time (in seconds) that the looped activity ran.

How to Test a Runbook

Once you build a runbook you can test it prior to having it run in production. To test, you use the **Runbook Tester** which is launched from the **Runbook Designer**. The **Runbook Tester** allows you to run the runbook viewing the published data from each activity. You can run through the entire runbook, step through each activity one at a time, or set breakpoints at certain activities.

▶ To test a runbook

1. In the **Runbook Designer**, open the runbook.
2. Click **Runbook Tester**.
3. If prompted, click **Yes** to check out the runbook.
4. To run through the runbook from beginning to end, without stopping click **Run to Breakpoint**.
If you want to step through it one activity at a time, click **Step**.
5. View the Log pane to see the completion status of each activity. View the details and published data from an activity by selecting it and clicking **Show Details**.

▶ To set a breakpoint

1. Select the activity to set the breakpoint on.
2. Click **Toggle Breakpoint**.
3. Click **Run to Breakpoint**.
Each activity up to the breakpoint runs. The runbook pauses before running the activity with the breakpoint.
4. Continue through to the end of the runbook by clicking **Run to Breakpoint** again or step through it one activity at a time by clicking **Step**.

Troubleshooting

Different logs are available that provide information about Orchestrator runbooks and servers. The logs that are available are listed in the following table with links to the sections where they are discussed in further detail:

Name	Contents	Where data is stored	Where data is viewed
Real time log	Live information about a running runbook instance	Orchestrator Database	Log tab in Runbook Designer
Historic Log	Historical information about instances of a runbook	Orchestrator Database	Log History tab in Runbook Designer
Runbook Audit	Audit information about	Orchestrator Database	Audit History tab in

Name	Contents	Where data is stored	Where data is viewed
History	changes to a runbook		Runbook Designer
Activity Events	Status information about Orchestrator Management server, Runbook servers, and database	Orchestrator 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 in 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, then this information will be included in the activity's detail.

Historic Log

You can view the historic log of a runbook in 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 is displayed 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, then this information will be included in the activity's detail.

Runbook log options

By default, the Runbook logs do not include the value of published data for each activity. You can change the logging properties for an individual runbook to include this information in the logs. There are two kinds of published data that you can configure separately.

Activity-specific Published Data is published data that is specific to a particular 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.

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

To configure runbook log options, see the section on logging in the [Runbook Properties](#) topic.



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, and 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 in the **Audit History** tab of the **Runbook Designer**. The entries for the currently selected runbook will be displayed. Double-click an entry to view the **Details** dialog box which will contain 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

If the number of log entries is allowed to grow indiscriminately, it can result in excessive storage requirements for the Orchestrator database. In order to maintain the health and performance of the database, you should regularly purge old log entries. This can be performed on either a scheduled or manual basis.

Automatically purging the Runbook logs

You can schedule the Runbook logs to be purged on a regular basis and select different options to specify the frequency of this process and how many entries to retain. By default, Runbook logs are automatically purged with the following settings.

Setting	Default Value
Frequency	Once per day
Time	1:00 AM
Retain	Keep last 10,000 entries

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

▶ **To change the automatic Runbook purge settings**

1. Right-click **Management server** and select **Log Purge**.
2. Ensure 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

There are two options to manually purge the Runbook logs as shown below. Both procedures are performed from the Runbook Designer.

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

1. Right-click **Management server** and 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 center pane, select the tab of the runbook.
3. In the **Log History** pane, select the **Log History** tab.
4. Click the Recycle Bin icon at the top of the pane.
5. When prompted if you're 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 center pane, select the tab of the runbook.
3. In the **Log History** pane, select the **Log History** tab.
4. Select a single log entry or use the Ctrl and Shift keys to select multiple entries.
5. Click the red Delete icon at the top of the pane.
6. When prompted if you're sure that you want to delete the selected entries, click **Yes**.

Removing orphaned log entries

Orphaned log files can be left in the Orchestrator 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 properly with its ending time. These entries will be removed when the log entries are purged using either of the methods described above. If you want to remove orphaned entries before the next time that the entries are purged, then you can manually run the **ClearOrphanedRunbookInstances** stored procedure to perform this function.

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

► To remove orphaned log entries

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



Note

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

4. Expand **Programmability** and then **Stored Procedures**.
5. Right click **Microsoft.SystemCenter.Orchestrator.Runtime.Internal.ClearOrphanedRunbookInstances** and 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 **Results** pane that the query has completed close **SQL Server Management Studio**.

Activity Events

Each activity in an Orchestrator runbook has the ability to send an event whenever it fails to run or taking too long to run. These events are presented in 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. Select the **Run Behavior** tab.
4. Type in 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 in the **Events** tab in the Runbook Designer, you can send them to an SNMP trap destination. This allows you to monitor the health of the Orchestrator environment using other tools designed to provide proactive alerting. The only requirement for such a tool is that it is able to 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 once for each destination that you want to add using the following syntax:

```
oedc /snmp /add /ip Target IP Address /port Target Port /version version /community community
```

For example, use the following procedure to send traps 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 privileges.
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, but you must instead 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 privileges.
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 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 will be 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

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. This enables you to report on configuration and change compliance of processes and provides a quick method to identify changes made to a third-party 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 may 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 privileges.
2. Navigate to *System Drive*:\Program Files (x86)\Microsoft System Center 2012\Orchestrator\Management Server.
3. To activate the Audit Trail, type **atlc /enable**
To deactivate the Audit Trail, type **atlc /disable**

Audit Trail Files

Audit Trail files are stored in csv format and have the details shown in the following table:

Log Type	File Name	Contents	Computer	Location
Runbook Publisher	<i>Computer Name</i> _RunbookPublisher_ <i>Timestamp</i> .csv	<ul style="list-style-type: none"> • Date and time that the runbook was started • User 	Management Server	<i>System Drive</i> :\ProgramData\Microsoft System Center 2012\Orchestrator\Audit\ManagementService
			Runbook Server	<i>System Drive</i> :\ProgramData\Microsoft System Center 2012\Orchestrator\Audit\RunbookService

Log Type	File Name	Contents	Computer	Location
		<p>name and domain that launched the runbook</p> <ul style="list-style-type: none"> • Name of the computer where the runbook ran 		
Activity Runtime Information	<i>Computer Name_ ObjectRuntimeInfo_Timestamp.csv</i>	<ul style="list-style-type: none"> • Date and time that 	Runbook Server	<i>System Drive:\ProgramData\Microsoft System Center 2012\Orchestrator\Audit\PolicyModule</i>

Log Type	File Name	Contents	Computer	Location
		<p>t act ivit y ran</p> <ul style="list-style-type: none"> • <p>Na me of run bo ok ser ver tha t ran the act ivit y</p> <ul style="list-style-type: none"> • <p>ID of the job pro ce ss tha t ran the act ivit y</p> <ul style="list-style-type: none"> • <p>Ob jec t XM L</p>		

Log Type	File Name	Contents	Computer	Location
		code that activity received as input data		

When a file reaches 200 megabytes in size, a new file is created. The timestamp is included in the file name in order 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 hidden.

Trace Logs

Orchestrator can create trace log messages on the Management server 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 hidden.

Modifying Trace Log Settings

Trace log settings are configured with registry values under the key **HKEY_LOCAL_MACHINE\SOFTWAREWow6432Node\Microsoft\SystemCenter2012\Orchestrator\TraceLogger**. There is a set of registry values for each of the Orchestrator services as shown in the following table:

Feature	Registry Key
Audit Trail utility	atlc.exe
Data Store Configuration utility	DBSetup.exe
Management Service	ManagementService.exe
Permissions Config utility	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 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 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 can be used 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.

Deploy and Launch Runbooks

This section describes how to deploy and manage runbooks in your environment.

Using runbooks

- [Deploying Runbooks](#)
- [Running Runbooks](#)
- [Runbook Server Properties](#)

Other resources for this product

- TechNet Library main page for **System Center Orchestrator 2012**
- [System Center Orchestrator 2012 Runbook Guide](#)
- [Runbook Concepts](#)
- [Tools](#)
- [Design and Build Runbooks](#)

Deploying Runbooks

Version control

In System Center Orchestrator 2012, multiple users can create and update runbooks. However, only one user at a time can make changes to a runbook. This protects your work from being overwritten by someone else with the same permission level.

To edit a runbook, you must check it out. Another user cannot edit that runbook until you either commit all changes by checking the runbook in or revert all changes by undoing the checkout.

Check In and Check Out

- **Check Out:** When a user is editing a runbook, the runbook is checked out and cannot be edited by anyone else. If someone else is already editing the runbook, a pop-up window opens informing you that someone is already editing the runbook.
- **Check In:** When a user editing the runbook performs a check-in operation, all changes that were made are committed, and other users can then edit the runbook after they

check it out. Check-in comments describe the changes that have been made.

- **Undo Check Out:** When a user editing the runbook performs an undo check out operation, all changes that were made are reverted after the runbook was checked out. After the undo check out operation is completed, another user can edit the runbook.

Audit Log

When a runbook has been changed and is checked in by a user, an entry appears in the **Audit History** log.



Tip

When a runbook has been altered to a state where it is no longer functioning, you can select the **Audit History** tab at the bottom of the Runbook Designer to see the changes that were made and then reverse them.

▶ To view runbook change details

1. From the Runbook Designer, select the **Audit History** tab at the bottom, double-click the entry item to open the **Details** dialog box.
2. From the **Name** column, click each item in the list to view the changes that were made.
3. When you select an item, the **Action** type will display beneath the **Activities** box. For example, **Action: Modified** or **Action: Added**. When you select the **Action: Modified** type, the **Attribute**, **Old Value**, and **New Value** will be listed in the bottom text box.

Running Runbooks

This topic describes the process for launching runbooks, viewing the results, and stopping a job.

▶ To launch a runbook

1. In the **Runbooks** pane, double-click the **New Runbooks** folder so you can see the **Monitor Runbook** and **Read Text File Runbook** in the folder tree.
In the **Details** pane, you should see **Summary**, **Runbooks**, **Jobs**, and **Instances** tabs.
2. In the **Details** pane, select the **Runbooks** tab.
3. In the bottom **Runbooks** pane, click **Monitor Runbook** or **Read Text File Runbook**.
4. In the **Actions** pane, select **Start Runbook**.
5. In the **Start Runbook** dialog box, go to **Available Runbook Server(s)** box and select the applicable server.
6. Click the arrow button so that the server name is now in the **Selected Runbook**

Servers(s) box.

7. Click **Start**.

If the runbook was run successfully, the **Orchestration Console** dialog box that opens will contain the following sentence.

`The operation was successfully completed.`

8. Click **OK**.

▶ To view runbook details

1. In the **Runbooks** pane, double-click the **New Runbooks** folder.
2. In the **Details** pane, click the **Runbooks** tab.
3. In the bottom **Runbooks** pane, click **Monitor Runbook** or **Read Text File Runbook**.
4. To view all jobs that have been created with your runbooks, in the **Actions** pane, select **View Jobs**.
5. To view all instances, in the **Actions** pane, select **View Instances**.
6. To view the activities in the runbook, in the **Actions** pane, select **View Definition**.

▶ To find events

1. Left-click **Start**, point to **All Programs**, click **Administrative Tools**, and then click **Event Viewer** to open a session.
2. On the Event Viewer menu, double-click **Windows Logs**, and then on the menu, click **Application**.

▶ To stop a job from the Runbook Designer

1. In **Runbook Designer**, click the **Monitor Runbook** tab.
2. On the toolbar, click **Stop**.

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 on the service account and how to change it, see the **Orchestrator Security Planning** topic in the System Center Orchestrator 2012 Deployment Guide.

▶ To view Runbook Server properties

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

Creating and Testing a Sample Runbook

The following section describes how to create and test a simple runbook. The purpose of this runbook is to detect when a text file is added to a particular folder, copy that file to another folder, read the contents of the file, append a line from the copied file to another file, and then delete the original file.

The runbook starts with a **Monitor File** activity to wait for the text file to be created. It then uses the **Copy File**, **Read Line**, **Append Line**, and **Delete File** activities to perform the other functions. A **Junction** activity is used to coordinate the activities so that the copy and append line are both completed before the source file is deleted.

Creating the runbook

Use the following procedures to create the runbook using the required activities.

▶ To create a runbook

1. Go to **Start -> All Programs -> Microsoft System Center 2012 -> Orchestrator-> Runbook Designer**.
2. In the **Connections** pane, right-click **Runbooks**, select **New**, and then select **Runbook..**
A new runbook tab appears at the top of the center workspace with the name **New Runbook**.
3. Right-click the **New Runbook** tab, and then click **Rename**.
Click the **Yes** button in the **Confirm Check out** dialog box.
4. Type **Append and Copy Workflow** in the input box and press Enter.
You have created a new runbook and are ready to begin adding and configuring activities.

▶ To add and configure a Monitor File activity

1. With the newly created **Append and Copy Workflow** runbook open, in the **Activity** pane expand the **File Management** category.
2. Click and drag the **Monitor File** activity to the Runbook Designer workspace.
3. Double-click the **Monitor File** activity to open its properties dialog box.
4. In the **In folder** box, type **C:\Drop**.
5. Click the **Add** button in the **Filters** section.
6. In the **Filter Settings** dialog box, in the **Name** dropdown, select **File Name**.
7. In the **Relation** dropdown, select **Matches Pattern**.
8. In the **Value** box, type ***.txt**.
9. Click **OK**.
10. Select the **Triggers** tab.

11. In the **Trigger if one of the files was** section, check the **Created** box, and then click **Finish**.

The **Monitor File** activity is created and configured to watch for any new text files that are created in the C:\Drop folder.

▶ **To add additional activities to the runbook**

1. In the **Activity** pane, expand the **File Management** category.
2. Click and drag the **Copy File** activity to the Runbook Designer.
3. Expand the **Text File Management** category.
4. Click and drag the **Read Line** activity to the Runbook Designer workspace.
5. Create a link between the **Monitor File** activity and the **Copy File** activity by clicking and dragging the right arrow of the **Monitor File** activity to the **Copy File** activity.
6. Create a link between the **Monitor File** activity and the **Read Line** activity by clicking and dragging the right arrow of the **Monitor File** activity to the **Read Line** activity.

By adding both the **Read Line** activity and the **Copy File** activity, you have created a workflow.

▶ **To configure the Copy File activity**

1. In the **Append and Copy Workflow** runbook right-click the **Copy File** activity, and then select **Properties**.
2. On the **Details** tab, right-click the **File** box, click **Subscribe**, and then click **Published Data** to open the **Published Data** dialog box.

The **Monitor File** activity is listed at the top of the **Published Data** dialog box because this is the activity just prior to the selected one.

3. From the **Name** column, select **Name and path of the file**, and then click **OK**. This populates the **File** property of the **Copy File** activity with the name and path of the file from the **Monitor File** activity.
4. In the destination **Folder** box, type **C:\Copy**.
5. Click **Finish**.

The **Copy File** activity is now configured to copy files from the target folder to the destination folder.

▶ **To configure the Read Line activity**

1. In the **Append and Copy Workflow** runbook, right-click the **Read Line** activity, and then select **Properties**.
2. On the **Details** tab, right-click the **File** box, click **Subscribe**, and then click **Published Data** to open the **Published Data** dialog box.
3. In the **Activity** dropdown, select **Monitor File**.
4. From the **Name** column, select **Name and path of the file**, and then click **OK**.
5. Click the ellipse button to the right of the **File encoding** box and select **auto**.

6. the **Line numbers** box, type **1-END**, and then click **OK**.
7. Click **Finish**.

The **Read Line** activity is now configured.

▶ To add an **Append Line** activity

1. In the **Activity** pane, expand the **Text File Management** category.
2. Click and drag the **Append Line** activity to the Runbook Designer workspace to the right of the **Read Line** activity.
3. Create a link from the **Read Line** activity to the **Append Line** activity by clicking and dragging the right arrow of the **Read Line** activity to the **Append Line** activity.
4. Right-click the **Append Line** activity, and then select **Properties**.
5. On the **Details** tab in the **File** box, type **C:\CopyMasterlog.txt**.
6. Click the ellipse button to the right of the **File encoding** box and select **auto**.
7. Right-click the **Text** box, click **Subscribe**, and then click **Published Data** to open the **Published Data** dialog box.
8. From the **Name** column for the **Read Line** activity, select **Line text**, and then click **OK**.
9. Click **Finish**.

The **Append File** activity is now configured to append files to the **Masterlog.txt** file.

▶ To synchronize branches of a runbook

1. In the **Activity** pane, expand the **Runbook Control** category.
2. Click and drag the **Junction** icon to the Runbook Designer workspace.
3. Create a link from the **Append Line** activity to the **Junction** activity by clicking and dragging the right arrow of the **Append Line** activity to the **Junction** activity.
4. Create a link from the **Copy File** activity to the **Junction** activity by clicking and dragging the right arrow of the **Copy File** activity to the **Junction** activity.
5. Right-click the **Junction** activity, then select **Properties**.
6. Click the ellipse button next to the **Return data from** box and select **Copy File**. Click **OK**. This action configures the activity to return the same published data as the **Copy File** activity.
7. Click **Finish**.

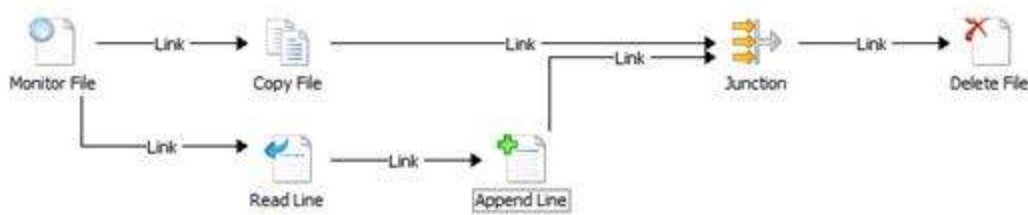
The **Junction** activity is configured to coordinate the workflow so that it no further activities run until both the **Copy File** activity and **Append Line** activity complete.

▶ To add and configure the **Delete File** activity

1. In the **Activity** pane, expand the **File Management** category.
2. Click and drag the **Delete File** icon to the design workspace.
3. Create a link from the **Junction** activity to the **Delete File** activity by clicking and dragging the right arrow of the **Junction** activity to the **Delete File** activity.

4. Right-click the **Delete File** activity, and then select **Properties**.
5. Right-click the **Path** box, click **Subscribe**, and then click **Published Data** to open the **Published Data** dialog box.
In the **Activity** dropdown, select **Copy File**.
6. From the **Name** column, select **Name and path of the origin file**, and then click **OK**.
7. Click **Finish**.

The **Append and Copy Workflow** runbook is now completed. It should look similar to the following image.



To test the runbook

You can test the runbook using the Runbook Tester. This tool allows you to run the entire runbook and inspect the completion status and output of each activity. The Runbook Tester runs the activities, so you must first create the folders specified for the runbook.



1. Create a folder on the runbook server called **C:\Drop**.
2. Create a folder on the runbook server called **C:\Copy**.
3. With the **Append and Copy Workflow** runbook selected in the Runbook Designer, click **Runbook Tester** on the toolbar.
4. Click **Run To Breakpoint**. The **Monitor File** activity is loaded and waits for a text file to be created in the **C:\Drop** folder.
5. Open **Notepad** and type a few lines of text. Save the file as **C:\Drop\File1.txt**.
6. Wait a few moments for the other activities to run. Ensure that each of the activities completed successfully.
7. To view the published data and other details of an activity, click **Show Details** for the activity.
8. Open the **c:\Drop** folder and ensure that the file has been removed.
9. Open the **C:\Copy** folder and ensure that the file has been copied. Also verify that the MasterLog.txt file has the contents of the original file.