Microsoft System Center 2012 R2

Using Runbooks in System Center 2012 R2 Orchestrator

Microsoft Corporation

Published: November 1, 2013

Applies To

System Center 2012 - Orchestrator

Orchestrator in System Center 2012 SP1

System Center 2012 R2 Orchestrator

Feedback

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

Copyright Information

Revision History

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

Contents

Using Runbooks in System Center 2012 - Orchestrator

Welcome to the System Center 2012 - Orchestrator Runbook Guide. This document describes Orchestrator runbooks. This document provides information about 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 2012 - Orchestrator.

For more information on building custom solutions with Orchestrator in System Center 2012, see System Center 2012 Integration Guide - Orchestrator.

Orchestrator Runbooks

- Runbook Concepts

  Provides basic information and terminology for runbooks, activities, and workflows.
- Tools

  Describes tools to build and start runbooks.
- Design and Build Runbooks

  Describes how to design and create a runbook.
- Deploy and Start Runbooks

  Describes how to deploy runbooks to your Orchestrator environment.
- Runbook Samples

  Provides samples and step-by-step guidance on how to build a runbook.

Other resources for this product

TechNet Library main page for **System Center Orchestrator 2012**

Getting Started with System Center 2012 - Orchestrator Release Candidate

Deploying System Center 2012 - Orchestrator Release Candidate

Administering System Center 2012 - Orchestrator Release Candidate

Integration Packs for System Center 2012 - Orchestrator Release Candidate

Using the Orchestration Console in System Center 2012 - Orchestrator Release Candidate

Runbook Activity Reference for System Center 2012 - Orchestrator Release Candidate

Runbook Concepts

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

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

Runbook Concepts

- Runbooks

  Provides configuration information for runbooks.
- Activities

  Describes categories of activities and their attributes.
- Workflow Control

  Describes tools to manage a runbook.

Other resources for this product

- TechNet Library main page for **Orchestrator for System Center 2012**
- Using Runbooks in System Center 2012 - Orchestrator
- Tools
- Design and Build Runbooks
- Deploy and Start Runbooks
- Runbook Samples

Runbooks

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

Runbook Attributes

- Runbook Properties

  Provides configuration information for individual runbooks.
- Runbook Permissions

  Describes access rights and permissions for single and multiple runbooks.

Other resources for this product

- TechNet Library main page for **System Center Orchestrator 2012**
- Using Runbooks in System Center 2012 - Orchestrator
- Runbook Concepts
- Activities
- Workflow Control

Runbook Properties

A runbook is essentially a series of activities that are using data, performing tasks, and publishing data for use by other activities in the runbook. Each runbook has a collection of configurable properties. These properties let you 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.
2. If the runbook is stored in a folder, select the appropriate folder under **Runbooks**.
3. In the **Runbook Designer** Design workspace, right-click the tab of a runbook to select **Properties**.
4. To close the **Runbook Properties** dialog box, click **Finish**.

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

General

On the **General** tab of the **Runbook Properties** dialog box, you can customize a name and description for the runbook. You can also associate a schedule with the runbook. After you assigned a schedule to the runbook, the runbook only runs on the dates and times that you 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 to select **New** to select**Schedule**.
3. On the **General** tab of the **New Schedule** dialog box, in the **Name** box, enter a name for the schedule.
4. On the **Details** tab of the **New Schedule** dialog box, select the date and time to start the runbook.

To configure the schedule for specific days of the week

a. On the **Details** tab of the **New Schedule** dialog box, click **Days of the week**, and then select the days on which you want to start the runbook.

b. Under **Occurrence**, select the week of the month to start the runbook.

For example, if you want to start 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 box, click **Days of the month**.

b. In the **Days of the month** box, enter the date or dates on which you want to start the runbook.

Separate multiple dates with a comma.

For example, if you want to start the runbook on the first and 15th of every month, enter **1, 15** in the **Days of the month** box.

To configure the schedule for specific hours

a. On the **Details** tab of the **New Schedule** dialog box, select **Hours**.
b. In the **Schedule Hours** dialog box, select the hours on which you want to start the runbook.

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

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

Important

The scheduled date and time to start a runbook is based on the system clock of the runbook server. This enables schedules to function in virtual machine environments and to continue to run even when the system clock adjusts for daylight savings time.

To associate a schedule with a runbook

1. In the **Runbook Properties** dialog box, on the **General** tab, click the ellipsis **(...)** button to browse for a **Schedule**.
2. Select a schedule, click **OK**, and then 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 found in the **Connections** pane of the Runbook Designer. If the runbook server that uses the Primary role is available, the runbook runs on it. If the primary runbook server is not available, each runbook server that uses a Standby role is checked until one is found 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 box, 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 that you added becomes the primary runbook server.
4. To add more 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 orchestration database. If stored in the orchestration database, this data is visible in views such as the **Log** pane in the Runbook Designer and in the Orchestration console. This information does not affect the availability of Published Data in a running runbook.

Published Data includes data specific to each activity. For detailed information about the Published Data available in each standard activity, see the **System Center Orchestrator 2012 Runbook Activity Reference**. For detailed information about published data available in integration packs, see **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

When you turn on logging, the size of the orchestration database increases.

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** box.

If you want to be notified if the runbook does not run, click the **Runbook fails to run** option.

For more information about Event Notifications, see **Orchestrator Logs**.

Job Concurrency

The job concurrency setting lets you set the maximum number of simultaneous jobs, so that you can carry out multiple requests for the same runbook at the same time. This setting applies to the individual runbook. A runbook server can run 50 runbooks at the same time. If you select a job concurrency setting over 50, your environment requires more runbook servers or the requests to start a runbook will queue.

The following limitations apply:

- You cannot run simultaneous requests for runbooks that start with Monitoring 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** value to 1 and displays an error message.
- A runbook server runs simultaneous requests for runbooks up to the maximum processing limit. To change the maximum processing limit, see **How to Configure Runbook Throttling**.
- Do not create simultaneous requests for runbooks that contain Modify Counter activities. When you run different copies of the same runbook at the same time 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 run at the same time.
- Do not run simultaneous requests for runbooks that interact with a non-Microsoft product, such as a ticketing or system-monitoring tool, unless you have a good understanding of how the tool handles parallel processing. If the non-Microsoft 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 can finish successfully. For example, if your runbook creates a folder, copies files into it, and then deletes the folder when it is finished, one instance of the runbook might 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 that a runbook returns when it finishes. 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 another runbook by 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.

See Also

System Center Orchestrator 2012 Runbook Activity Reference

Troubleshooting

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** Design workspace, right-click the tab for a runbook to 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** check box next to **Read**.

   If the user or security group should be able to change the runbook, select the **Allow** check box next to **Write**.

   If the user or security group should be able to change permissions for the runbook, select the **Allow** check box next to **Full Control**.
5. To close the **Permissions for Runbook** dialog box and save any changes, click **OK**.

See Also

Runbook Concepts

Activities

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

- Access Published Data
- Perform some action
- 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

  Describes standard activities available in Orchestrator.
- Monitoring Activities

  Describes specialized activities that monitor environment states and event logs.
- Customized Activities

  Describes customization options available in Orchestrator.
- Common Activity Properties

  Describes configurable properties common to all activities.

Other resources for this product

- TechNet Library main page for **System Center Orchestrator 2012**
- Using Runbooks in System Center 2012 - Orchestrator
- Runbook Concepts
- Runbooks
- Workflow Control

Standard Activities

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

For detailed reference information about each standard activity, see **Standard Activities** in the**Runbook Activity Reference for System Center 2012 - Orchestrator Release Candidate**.

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

Activity collection	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.

Activity collection	Description
Text File Management	Manipulates text files.
Utilities	Supports working and manipulating data within a workflow.

See Also

Activities

Monitoring Activities

Monitoring activities are a specialized group of activities that are triggered by a state or event of a task 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 run repeatedly on a predefined schedule.

An Orchestrator activity cannot trigger a monitoring activity. A monitoring activity is a start condition within a runbook. The Monitor Folder activity waits for the files within a specified folder to change. When a file changes, the Monitor Folder activity triggers the next activity in the runbook workflow.

Runbooks that start with a monitoring activity load the monitoring activity and wait for the trigger condition to occur. When the monitor activity 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 the Orchestration console.

For a list of standard monitoring activities, see **Monitoring Activities** in the **Runbook Activity Reference for System Center 2012 - Orchestrator Release Candidate**.

See Also

Monitoring Activities

Runbook Activity Reference for System Center 2012 - Orchestrator Release Candidate

Customized Activities

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

See Also

Activities

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

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 for each activity in the **System Center Orchestrator 2012 Runbook Activity Reference**.

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.

Published Data Behavior

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

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 comma-separated value format (.csv file), which is useful for importing into spreadsheets or databases.

Event Notifications

Some activities are expected to take a limited amount of time to finish. If the activity does not finish within the specified period, the activity can be stalled or another issue prevents the activity from finishing. 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 about event notifications, see **Troubleshooting**.

Event notification setting	Description
Report when the activity runs for more than	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 a run failure notification.

Security Credentials

The settings on the Security Credentials tab let you specify the account that runs the Runbook Server Service. This is useful when the activity performs activities with resources on a remote computer.

Important
Note that the account used to start the runbook must have permission on the local computer to run successfully.

Important
If you use the **Invoke Runbook** activity and you modify Security Credentials, the account you use must be a member of the Orchestrator System group to run successfully.

Caution
If permissions on the Orchestrator installation path are changed and the activity's Security Credentials has a custom user account that does not include **Read/Execute** permissions to **ExecutionData.dll** on the runbook server, the activity will fail.

Option	Behavior
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 credentials to perform this action. If the credentials you provided fail validation, the account assigned to the runbook server account is used.

See Also

Orchestrator Security Planning

System Center Orchestrator 2012 Runbook Activity Reference

Workflow Control

When you build runbooks in System Center 2012 - Orchestrator, 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

The workflow control provides the following controls.

- Starting Point
- Smart Links
- Embedded Loops

Other resources for this product

- TechNet Library main page for **System Center Orchestrator 2012**
- Using Runbooks in System Center 2012 - Orchestrator
- 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 completed.

If a runbook starts with any activity other than a monitor activity, the runbook begins processing and attempts to run to completion. If the runbook starts with a monitoring activity, the monitor loads and waits for the trigger condition. When the condition is met, a runbook instance is created to run the remaining activities in the runbook. The monitor continues to run and waits for 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

Monitoring Activities

Smart Links

The links that connect individual activities in a runbook are called smart links. Smart links in System Center 2012 - Orchestrator support precedence between two activities. Smart links invoke the next activity in the runbook as soon as the previous activity finishes 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** Design workspace, click and drag two activities from the **Activities** pane to the **Runbook Designer** Design workspace.
2. In the **Runbook Designer** Design 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

- To disable the smart link, right-click the smart link to toggle **Enable**.

  The smart link changes to a dashed line indicating that it is disabled.

To enable a smart link connection

- To enable the smart link, right-click a disabled smart link to toggle **Enable**.

  The smart link changes to a solid line to indicate that it is enabled.

General Tab

In the **Link Properties** dialog box, on the **General** tab, 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

- To view the smart link name, right-click a smart link to select **Properties**.
- In the **Properties** dialog box, on the **General** tab, in the **Name** box, enter a descriptive name.
- Click **Finish**.

To display smart link names in the runbook

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

Include and Exclude Tabs

Orchestrator lets you configure conditions for passing data to the following tasks in the runbook. By using link conditions, you can 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.

On the **Include** tab, you can 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.

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 to select **Properties** to open the **Link Properties** dialog box.

   Important

   To change the values that make up the rule, you have 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 then either select or type in an appropriate value. For more information about criteria, see Smart Link Criteria.
6. Click **Finish**.

To remove a smart link condition

1. In the **Link Properties** dialog box, click either the **Include** tab or **Exclude** tab.
2. To select the condition that 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

In the **Link Properties** dialog box, on the **Options** tab, 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 that you want.
2. Click **Width** of the smart link line in pixels to specify the width.
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 depends 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 status returns one of the following values:

- success
- warning
- failed

Each time you create a new link, it creates 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 finished, you should delete or change criteria.

Binary Values

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

Text Values

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

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.

Condition	Description
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$** returns a false, but **.*text$** returns true. Similarly, **^This** returns false, but **^This.*** returns **true**.

Numeric Values

Certain published data properties return numeric data that you can compare to an expected value. The following table shows the different criteria that can be used.

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 then	The value of the Published Data item is less than the specified value.
is greater then	The value of the Published Data item is greater than the specified value.

Condition	Description
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.

See Also

[Workflow Control]()

Embedded Loops

In System Center 2012 - Orchestrator, 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 as 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** that let you use information from the loop itself in the looping conditions.

Loops run one time for each incoming piece of data that is 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, the next item goes through **Append Line** and loops until it exits, and then the third begins. After all three items have been processed, 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 to select **Looping**. The **Looping Properties** dialog box opens.
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 Conditions

The rules on the **Exit** tab specify the conditions that determine whether the loop exits. The rules on the **Do Not Exit** tab specify 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. In the **Looping Properties** dialog box, click 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 have 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 then 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 about criteria, see **Smart Link Criteria**.
6. Click **Finish**.

To remove an exit condition

1. In the **Looping Properties** dialog box, click either the **Exit** tab or the **Do Not Exit** tab.
2. To select the condition you want to remove, click **Or** to the right of the link condition, and then click **Remove**.
3. Click **Finish**.

See Also

Workflow Control

Tools

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

- Runbook Designer

  Create, manage, and run runbooks.
- Runbook Tester

  Step through a runbook to test its functionality.

Other resources for this product

- TechNet Library main page for **System Center Orchestrator 2012**
- Using Runbooks in System Center 2012 - Orchestrator
- Runbook Concepts
- Design and Build Runbooks
- Deploy and Start Runbooks
- Runbook Samples

Runbook Designer

The **Runbook Designer** is the tool that you use to create, manage, and run runbooks in System Center 2012 - Orchestrator. The **Runbook Designer** is intended for users who must create or modify runbooks. Users who only have to run runbooks and view their status should use the Orchestration console which is documented in the **System Center 2012 - Orchestrator 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**.
Runbook Designer workspace	The workspace where you build Orchestrator runbooks. The runbooks in the folder selection in the **Connections** pane are listed as tabs across the top of the workspace. When you select a tab in a runbook, it is displayed in the Runbook Designer workspace.
Activities	Contains all the activities available (either standard activities or activities available from integration packs) for use in runbooks. You drag activities from the **Activities** pane into the Design workspace, and then link them together

Pane	Description
	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 2012 - Orchestrator lets you sort activities alphabetically by activity name, or by category name. By default, activities are sorted by category, such as Runbook Control, Email, File Management, Monitoring, Notification, Scheduling, System, Text File Management, and Tools.

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

To sort activities alphabetically by activity name

- In the **Activities** pane, right-click a category name to select **All Activities**.

  The activities are sorted alphabetically by activity name.

To sort activities alphabetically by category name

- In the **Activities** pane, right-click a category name to 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

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

See Also

Tools

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 start Runbook Tester from the toolbar above the central Design workspace in the **Runbook Designer**. When you click the Runbook Tester button, the Runbook

Tester starts and loads 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 Microsoft SQL Server.

Important
Note that the account used to start the runbook must have permission on the local computer to run successfully. These permission requirements also apply when testing the runbook with the Runbook Tester. To successfully test your runbook, start the Runbook Designer **as Administrator**. By association, the Runbook Tester runs **as Administrator** and uses the higher-level security token.

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 each activity to view its information in the

Pane	Description
	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.

See Also

Tools

Design and Build Runbooks

This section provides details about how to design, build, and test, runbooks by using System Center 2012 - Orchestrator.

Runbooks

- Designing a Runbook

  Provides design guidance for building runbooks.
- Building a Runbook

  Describes how to create a runbook, how to add and link activities, and how to configure runbook properties.
- How to Test a Runbook

  Describes how to test a runbook.

Other resources for this product

- TechNet Library main page for **System Center Orchestrator 2012**
- Using Runbooks in System Center 2012 - Orchestrator
- Runbook Concepts
- Tools
- Deploy and Start Runbooks
- Runbook Samples

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 their 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.

See Also

Design and Build Runbooks

Building a Runbook

This topic describes the basic process for building a System Center 2012 - Orchestrator runbook.

Note

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

Step	Description
1. Create a runbook.	Create an empty runbook in the Runbook Designer.
2. Add activities.	Click and drag activities from the **Activities** 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 the runbook.	Save your changes and check in the runbook.

To create a new runbook

1. On the computer where the Runbook Designer is installed, click **Start**, point to **All Programs**, click **System Center 2012 - Orchestrator**, and then click **Runbook Designer**.
2. In Runbook Designer, in the **Connections** pane, click the **Runbooks** folder.
3. In the **Connections** pane, click the **Create a new runbook** icon.
4. In the **Runbook Designer** Design 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, drag an activity to the Design workspace 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 to select **Properties**. The **Runbook Properties** dialog box opens.
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

- In 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**
- Using Runbooks in System Center 2012 - Orchestrator
- Designing a Runbook
- How to Test a Runbook

Data Manipulation

With System Center 2012 - Orchestrator, 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**
- Using Runbooks in System Center 2012 - Orchestrator
- Building a Runbook
- Published Data

Computer Groups

System Center 2012 - Orchestrator is designed to interact with all of your data center systems. Computer groups let you 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.

You can create computer groups by using Active Directory queries, and 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 instances of Microsoft SQL Server, when an instance of SQL Server is added to your Active Directory system, it is automatically 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 also organize your computer groups into folders. Use the following steps to create a new folder.

To create a folder

1. In the **Connections** pane in the Runbook Designer, click the **Computer Groups** folder or a subfolder.
2. Right-click to select **New**, and then click **Folder**.

Use the following procedure to add a computer group. To add computers by using an Active Directory query or a System Center 2012 Configuration Manager collection, use the Active Directory Integration Pack or the Integration Pack for System Center 2012 Configuration Manager.

To add a computer group

1. In the **Connections** pane, right-click the **Computer Groups** folder or a subfolder.
2. Select **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, in the **Name** and **Description** boxes, 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 **Add Computer to Computer Group** dialog box.

6. Enter 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.
7. To add more computers to the group. repeat the previous two steps.

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** on 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.

See Also

Data Manipulation

Counters

When building runbooks in System Center 2012 - Orchestrator, 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 let you modify and check the status of a number that you can use to keep track of important statistics. You create a counter in the Connections pane in the Runbook Designer, and then get and modify it by 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.

Security

The access permissions for counters can be modified, but the Runbook server does not enforce these permissions.

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.

Important

Orchestrator does not support moving multiple counters with multiple-selection. To move more than one counter to another folder, you must move each counter individually.

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

To create a counter

1. In the **Connections** pane, double-click the **Global Settings** folder, right-click the **Counters** folder or a subfolder of the **Counters** folder to select **New**, and then click **Counter** to open the **New Counter** dialog box.
2. In the **Name** box, type a name for the counter.
3. In the **Description** box, type a description that explains the purpose of the counter.
4. In the **Default Value** box, type the starting 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 to select **Delete**.
6. Click **Finish**.

To organize counters

1. You can group counters in folders to organize them. To create a folder, right-click the **Counters** folder to select **New**, and then click **Folder**.
2. To move a counter to a different folder, right-click the counter to select **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.

See Also

Get Counter Value

Modify Counter

Published Data

Functions

By using System Center 2012 - Orchestrator, you can manipulate string data from text files, Published 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 lets you 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	Field('John;Smith;9055552211', ';', 2) returns 'Smith'

Function and Definition	Usage	Parameters	Example
		(starting at 1).	
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 comma (,).	Sum(2,3,4,5) returns '14'
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	Div(8, 4) returns '2' Div(9, 2, 2) returns '4.50'

Function and Definition	Usage	Parameters	Example
		decimal places that the result will be rounded to.	
Instr - returns the position of first occurrence of text within another text.	Instr ('SearchText', 'TextToFind')	SearchText - the text that is being searched. TextToFind - the text that you are searching for.	Instr('This is a string that is searched', 'string') returns 11
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	Mid('Take from the middle', 5, 4) returns 'from'

Function and Definition	Usage	Parameters	Example
		returned.	
LTrim - trims leading spaces from text.	LTrim('Text')	Text - the text that is 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 that is being trimmed of trailing spaces.	RTrim(' Remove the trailing spaces only. ') returns ' Remove the trailing spaces only.'
Trim - trims leading and trailing spaces from text.	Trim('Text')	Text - the text that is being trimmed.	Trim(' Remove leading and trailing spaces. ') returns 'Remove leading and trailing spaces.'
Len - returns the length of text.	Len('Text')	Text - the text that is being measured.	Len('Measure this text') returns 17

See Also

Data Manipulation

Regular Expressions

In System Center 2012 - Orchestrator, regular expressions let you 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.*"

Character	Meaning
	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 "abbbbb", "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.
$	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] ensures that none of the letters in the alphabet are present.

Character	Meaning
( )	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".
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.

See Also

Data Manipulation

Schedules

System Center 2012 - Orchestrator 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 runs 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.

Security

> The access permissions for schedules can be modified, but the runbook server does not enforce these permissions.

Note

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.

Note

Orchestrator does not support moving multiple schedules with multiple-selection. To move more than one schedule to another folder, you must move each schedule individually.

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 checks a particular schedule and returns 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. In 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, in the **Name** box, type a name for the schedule.
3. In the **Description** box, 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** includes the first and third day of the month. Typing **1-21** includes 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

Check Schedule

Variables

When building runbooks in System Center 2012 - Orchestrator, some settings are the same across activities. Variables let you specify a value that activities use in any runbook.

Security

The access permissions for variables can be modified, but the runbook server does not enforce these permissions.

Important

Be aware that in Orchestrator, variables that reference system variables, for example **%ProgramFiles%**, return values from a 32-bit runtime environment. This is because Orchestrator is a 32-bit application.

Note

Orchestrator does not support moving multiple variables with multiple-selection. To move more than one variable to another folder, you must move each variable individually.

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

To create a variable

1. In the **Connections** pane in the Runbook Designer, expand the **Global Settings** folder, and then click the **Variables** folder.
2. Right-click the **Variables** folder or a subfolder of the **Variables** folder to select **New**, and then click **Variable** to open the **New Variable** dialog box.
3. In the **Name** box, type a name for the variable.
4. In the **Description** box, type a description that explains the purpose of the variable.
5. In the **Value** box, type the value of the variable. This value replaces the placeholder in those activities where the variable is inserted.
6. If you want the variable to be encrypted (for example, to store a password for use in other runbook activities), select the **Encrypted Variable** check box.

   For more information about best practices for using encrypted variables, see **Orchestrator Data Encryption**.
7. Click **Finish**.

Important

System Center 2012 - Orchestrator does not let you combine an encrypted variable with plain text as a parameter value in a runbook.

To insert a variable in an activity

1. Right-click the applicable activity from your runbook to select **Properties**, and then click the **Details** tab to open the activities properties dialog box.
2. In a text box, to open a menu, right-click to select **Subscribe**, and then click **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 to select **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%: This variable 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, the text '%ENVVAR%' is returned).

See Also

Data Manipulation

Published Data

Published Data lets an activity use information from 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 that provide 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 has 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 describes the categories of Published Data value types.

Published Data value type	Description
String value	Text, for example, an error message 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.

Date and time characteristics

Activity Start Time and Activity End Time data is saved to the databus in two fields. These formats are local time and Coordinated Universal Time (UTC), both in ISO 8601 format. By using UTC, runbooks can run in either a non-locale-specific or a non-time-zone-specific context. Only Published Data that is saved to the databus provide date and time information in UTC with ISO 8601 formatting.

The date and time values displayed in the Runbook Designer and the Orchestration console, including, but not limited to the Log History, Audit History, and Events respect the locale date and time format configured for your computer.

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. The **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 Returned Data to Activity Configurations

When an activity has subscribed to Published Data, a placeholder is inserted where the value of the data will be 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 to select **Properties**, and then click the **Details** tab to open the activity's properties dialog box.
2. To open a menu, in the text box, right-click to select **Subscribe**, and then click **Published Data** to open the **Published Data** dialog box.
3. In the **Activity** 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. In the text box, click the data placeholder to open the **Published Data** dialog box.
2. In the **Activity** 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 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 have to modify text from a **Published Data** activity before you use it in another activity. For example, you might have 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 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 that the activity returns.
Runbook name	The name of the runbook.
Runbook Process ID	The process ID of the runbook module's executable program that is running on the runbook server. 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.
Server name	The name of the runbook server where the runbook is running.
Activity duration	The total time that the 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.

Name	**Description**
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 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.

Name	Description
	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.

See Also

Building a Runbook

How to Test a Runbook

After you build a runbook, you can test it before it is run in production. To test, you use the **Runbook Tester** which you start in the **Runbook Designer**. The **Runbook Tester** lets you run the runbook to view 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, and on the menu bar, click **Runbook Tester**.
2. If prompted, click **Yes** to check out the runbook.
3. 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**.
4. View the **Log** pane to see the completion status of each activity. To view the details and Published Data from an activity, select the activity, and click **Show Details**.

To set a breakpoint

1. Select the activity on which to set the breakpoint.
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. To continue through to the end of the runbook, click **Run to Breakpoint** again, or to step through it one activity at a time, click **Step**.

See Also

Design and Build Runbooks

Deploy and Start Runbooks

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

Using runbooks

- Deploying Runbooks
- Running Runbooks

Other resources for this product

- TechNet Library main page for **System Center Orchestrator 2012**
- Using Runbooks in System Center 2012 - Orchestrator
- Runbook Concepts
- Tools
- Design and Build Runbooks
- Runbook Samples

Deploying Runbooks

There are tools available in Orchestrator to help you manage the versions of your runbooks. These tools are described in the following sections.

Version Control

In System Center 2012 - Orchestrator, 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. In the Runbook Designer, select the **Audit History** tab at the bottom, double-click the entry item to open the **Details** dialog box.
2. In 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 displays 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** are listed in the bottom text box.

See Also

Deploy and Start Runbooks

Running Runbooks

This topic describes the process for starting runbooks, viewing the results, and stopping a job in the Runbook Designer.

To start a runbook

1. In the Runbook Designer, in the **Connections** pane, click the **Runbooks** folder to see the available runbooks.
2. In the Design workspace, click a runbook tab.
3. If the runbook is Checked Out, select the **Check In** button.
4. In the Design workspace, right-click the runbook tab and select **Run**.
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**.

To find events

1. 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**.

See Also

Deploy and Start Runbooks

Runbook Samples

This topic provides instructions about how to build and test sample runbooks by using the **Standard Activities** found in System Center 2012 - Orchestrator.

Samples

- Creating and Testing a Sample Runbook

  Provides step-by-step instructions about creating and testing a simple runbook.

- Monitor a Folder within a Runbook

  Provides instructions about how to create a runbook that monitors the activity in a folder.

Other resources for this product

- TechNet Library main page for **System Center Orchestrator 2012**
- Using Runbooks in System Center 2012 - Orchestrator
- Runbook Concepts
- Tools
- Design and Build Runbooks
- Deploying Runbooks

Creating and Testing a Sample Runbook

The following topic 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 File** and **Append Line** activities are both completed before the source file is deleted.

Creating the runbook

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

To create a runbook

1. Click **Start**, point to **All Programs**, click **Microsoft System Center 2012**, click **Orchestrator**, and then click **Runbook Designer**.
2. In the **Connections** pane, right-click **Runbooks** to select **New**, and then click **Runbook**.

   A **New Runbook** tab appears at the top of the **Runbook Designer** Design workspace with the name **New Runbook**.
3. Right-click the **New Runbook** tab to select **Rename**.

   In the **Confirm Check out** dialog box, click **Yes**.
4. Type **Append and Copy Workflow** in the **Input** box, and then 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 **Activities** pane, expand the **File Management** category.
2. Click and drag the **Monitor File** activity to the **Runbook Designer** Design workspace.
3. Double-click the **Monitor File** activity to open its **Properties** dialog box.
4. In the **In folder** box, type **C:\Drop**.
5. In the **Filters** section, click the **Add** button.
6. In the **Filter Settings** dialog box, in the **Name** list, select **File Name**.
7. In the **Relation** list, select **Matches Pattern**.
8. In the **Value** box, type ***.txt**.
9. Click **OK**.
10. Click the **Triggers** tab.
11. In the **Trigger if one of the files was** section, select the **Created** check 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 **Activities** pane, expand the **File Management** category.
2. Click and drag the **Copy File** activity to the **Runbook Designer** Design workspace.
3. Expand the **Text File Management** category.

4. Click and drag the **Read Line** activity to the **Runbook Designer** Design workspace.
5. To create a link between the **Monitor File** activity and the **Copy File** activity, click and drag the right arrow of the **Monitor File** activity to the **Copy File** activity.
6. To create a link between the **Monitor File** activity and the **Read Line** activity, click and drag 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 to select **Properties**.
2. On the **Details** tab, right-click the **File** box to select **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 before to the selected activity.
3. In 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 of and path to 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 source 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 to select **Properties**.
2. On the **Details** tab, right-click the **File** box to select **Subscribe**, and then click **Published Data** to open the **Published Data** dialog box.
3. In the **Activities** list, select **Monitor File**.
4. In 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 then select **auto**.
6. In 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 **Activities** pane, expand the **Text File Management** category.
2. Click and drag the **Append Line** activity to the **Runbook Designer** Design workspace to the right of the **Read Line** activity.
3. To create a link from the **Read Line** activity to the **Append Line** activity, click and drag

the right arrow of the **Read Line** activity to the **Append Line** activity.

4. Right-click the **Append Line** activity to select **Properties**.
5. On the **Details** tab in the **File** box, type **C:\Copy\Masterlog.txt**.
6. Click the ellipse (**…**) button to the right of the **File encoding** box, and then select **auto.**
7. Right-click the **Text** box to select **Subscribe**, and then click **Published Data** to open the **Published Data** dialog box.
8. In 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 **Activities** pane, expand the **Runbook Control** category.
2. Click and drag the **Junction** icon to the **Runbook Designer** Design workspace**.**
3. To create a link from the **Append Line** activity to the **Junction** activity, click and drag the right arrow of the **Append Line** activity to the **Junction** activity.
4. To create a link from the **Copy File** activity to the **Junction** activity, click and drag the right arrow of the **Copy File** activity to the **Junction** activity.
5. Right-click the **Junction** activity to select **Properties**.
6. Click the ellipse (**…**) button next to the **Return data from** box, and then select **Copy File**. Click **OK**. This action configures the activity to return the same Published Data as the **Copy File** activity.
7. Click **Finish**.

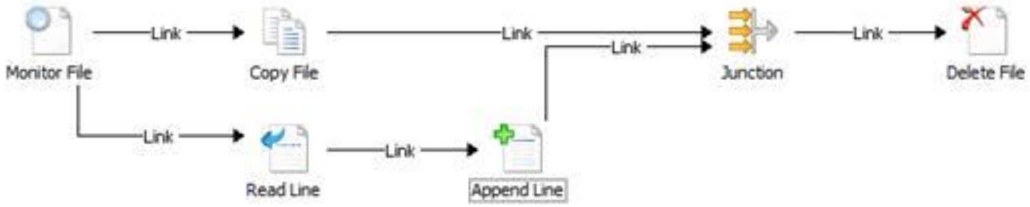
The **Junction** activity is configured to coordinate the workflow so that no further activities run until both the **Copy File** activity and **Append Line** activity finish.

To add and configure the Delete File activity

1. In the **Activities** pane, expand the **File Management** category.
2. Click and drag the **Delete File** icon to the **Runbook Designer** Design workspace.
3. To create a link from the **Junction** activity to the **Delete File** activity, click and drag the right arrow of the **Junction** activity to the **Delete File** activity.
4. Right-click the **Delete File** activity to select **Properties**.
5. Right-click the **Path** box to select **Subscribe**, and then click **Published Data** to open the **Published Data** dialog box.

   In the **Activity** list, select **Copy File**.
6. In the **Name** column, select **Name and path of the original file**, and then click **OK**.
7. Click **Finish**.

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



Testing the runbook

You can test the runbook by using the Runbook Tester. This tool lets you 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.

To test the runbook

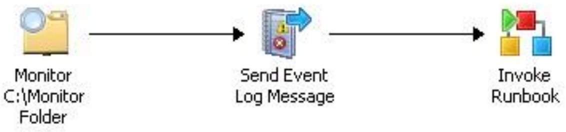
1. Create a folder on the runbook server called C:\Drop.
2. Create a folder on the runbook server called C:\Copy.
3. With the **Append and Copy Workflow** runbook selected in the Runbook Designer, on the toolbar, click **Runbook Tester**.
4. Click **Run To Breakpoint**. The **Monitor File** activity 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 is 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.

See Also

Runbook Samples

Monitor a Folder within a Runbook

This sample shows you how to create a simple monitor runbook that monitors a folder for new text files. When a file is detected, the runbook sends an event log message, and then starts another runbook.



Create and test a monitor runbook

The procedures to create, configure, and test a sample runbook that monitors a folder are described below.

To create the workflow

1. In the Runbook Designer **Connections** pane, right-click the **Runbooks** folder to select **New**, and then click **Runbook**.
2. Right-click the **New Runbook** tab to select **Rename**.
3. In the **Confirm Check out** dialog box, click **Yes**.
4. Type a name for the runbook, such as **Monitor Runbook**, and then press Enter.
5. In the **Activities** pane, click **File Management** to expand the category, and then drag the **Monitor Folder** activity into the **Runbook Designer** Design workspace.
6. In the **Activities** pane, click **Notification** to expand the category, and then drag the **Send Event Log Message** activity into the **Runbook Designer** Design workspace, to the right of the **Monitor Folder** activity.
7. In the **Runbook Designer** Design workspace, move your pointer over the right side of the **Monitor Folder** activity to display the smart link arrow.
8. Click the smart link arrow, and then drag it to the **Send Event Log Message** activity.
9. In the **Activities** pane, click **Runbook Control** to expand the category, and then drag the **Invoke Runbook** activity into the **Runbook Designer** Design workspace, to the right of the **Send Event Log Message** activity.
10. In the **Runbook Designer** Design workspace, move your pointer over the right side of the **Send Event Log Message** activity to display the smart link arrow.
11. Click the smart link arrow, and then drag it to the **Invoke Runbook** activity.

To configure the workflow

1. In the **Runbook Designer** Design workspace, double-click the **Monitor Folder** activity.
2. In the **Monitor Folder Properties** dialog box, click the **General** tab.
3. In the **Name** box, change the name of the activity to something informative, for example **Monitor C:\Monitor Folder**.
4. Click the **Details** tab.
5. On the **Details** tab, in the **Path** box, type the path of the folder you want to monitor, for

example **C:\Monitor**.

6. Below the **File Filters** list, click **Add**.
7. In the **Filter Settings** dialog box, set the following:

   a. In the **Name** list box, select **File Name**.

   b. In the **Relation** list box, select **Matches pattern**.

   c. In the **Value** box, type ***.txt**.

   This setting directs the monitor to look for files with the **txt** extension. This field accepts regular expression syntax.

8. Click **OK**.
9. Select the **Triggers** tab.
10. Select the **Number of files is** option, set the value in the list to **greater than**, and then type **0** in the edit box.
11. Click **Finish**.
12. In the **Runbook Designer** Design workspace, double-click the **Send Event Log Message**.
13. In the **Send Event Log Message Properties** dialog box, on the **Details** tab, in the **Properties** section, set the following:

    a. In the **Computer** box, type the name of the computer to receive the Event message.

    This is typically the computer where you are running Runbook Designer.

    b. In the **Message** box, type the message to display in the Event log, for example, **File Detected**.

    c. Leave the **Severity** level at **Information**.

14. Click **Finish**.

    Note

    In this sample, the **Invoke Runbook** activity is not configured. For more information about configuring this activity, see the **Invoke Runbook** activity in the System Center 2012 - Orchestrator Runbook Activity Reference.

To modify runbook settings

1. Above the **Runbook Designer** Design workspace, right-click the **Monitor Runbook** tab to select **Properties**.
2. In the **Monitor Runbook Properties** dialog, click the **Logging** tab, and then select both **Store Activity-specific Returned Data** and **Store Common Returned Data**.

3. Click **Finish**.
4. Right-click the **Monitor Runbook** tab to select **Check In**.

Test the runbook

In the Runbook Tester, you can test runbooks in a simulated runtime and 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 that you select.

Use the following steps to test your runbook in the **Runbook Tester**.

To prepare your computer

1. Right-click **Start** to select **Open Windows Explorer**.
2. Create a **C:\Monitor** folder on your computer.
3. Create a **C:\Source** folder on your computer.
4. In the **C:\Source** folder, create a file with a **txt** extension, for example **text.txt**.

To test the runbook

1. In the **Runbook Designer** Design workspace, select the **Monitor Runbook** tab.
2. On the toolbar above the **Runbook Designer** Design workspace, click **Runbook Tester**.
3. In the **Confirm Check out** dialog box, click **Yes**.
4. In **Runbook Tester**, on the toolbar, click **Step Over** to start stepping through the runbook.

   Tip
   To increase the size of the **Log** pane, remove the **Resource Browser** pane by selecting **View** on the menu, and then clearing the **Resource Browser** option.

5. In Windows Explorer, browse to the **C:\Source** folder.
6. Copy **test.txt** to **C:\Monitor**.
7. Close Windows Explorer.
8. On the Runbook Tester toolbar, click **Next**.

   After a few moments, note that the **Log** pane entry updates and shows an event for the **Monitor Folder** activity.

9. On the **Log** pane Click the **Show Details** link to see the contents of the data bus for that runbook.
10. Scroll down the list of properties. Note that the activity status is **success** indicating that the **Monitor Folder** activity detected the change in the folder.
11. On the Runbook Tester toolbar, click **Next.**

    Notice that the **Log** pane changes and shows an event for the **Send Event Log Message** activity.

12. Click the **Show Details** link and note that the activity status is **success** indicating that the

Send Event Log Message activity detected the change in the folder.

13. Close **Runbook Tester**.
14. On the **Runbook Designer** toolbar, click **Check In**.

See Also

Runbook Samples