

Microsoft System Center 2012 R2

Getting Started With System Center 2012 R2 Orchestrator

Microsoft Corporation

Published: November 1, 2013

Applies To

System Center 2012 Service Pack 1 (SP1)

System Center 2012 R2 Orchestrator

Feedback

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

Copyright Information

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

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

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

© 2013 Microsoft Corporation. All rights reserved.

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

Revision History

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

Contents

Getting Started with System Center 2012 - Orchestrator	4
What's New in System Center 2012 R2 Orchestrator	4
What's New in Orchestrator in System Center 2012 Service Pack 1	5
Orchestrator Capabilities	6
Orchestrator Architecture	8
Orchestrator Terminology	12
Glossary for System Center 2012 - Orchestrator	12
Glossary for Opalis Integration Server 6.3	14
Orchestrator Resources	16

Getting Started with System Center 2012 - Orchestrator

Orchestrator provides a workflow management solution for the data center. Orchestrator lets you automate the creation, monitoring, and deployment of resources in your environment. This guide describes the architecture of Orchestrator and includes definitions of key terms and concepts and information about where to find additional resources. After reading this guide, you should have a basic understanding of how Orchestrator works and where you can find more information.

Getting Started topics

- [What's New in System Center 2012 R2 Orchestrator](#)
Provides information about new features and integration packs for System Center 2012 R2 Orchestrator.
- [What's New in Orchestrator in System Center 2012 Service Pack 1](#)
Provides information about new features and integration packs for Orchestrator.
- [Orchestrator Capabilities](#)
Describes the features available in Orchestrator.
- [Orchestrator Architecture](#)
Describes the architecture of a basic Orchestrator deployment.
- [Orchestrator Resources](#)
Provides additional resources to help you use Orchestrator.

Other resources for this product

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

What's New in System Center 2012 R2 Orchestrator

The following new features are available in System Center 2012 R2 Orchestrator.

What's New

The following new features are added in this release:

- You can install the Service Management Automation web service and up to three runbook workers from System Center 2012 R2 Orchestrator Setup program. These can be used as part of the Windows Azure Pack for Windows Server configuration or to enable you to run runbooks and perform other automation tasks using Windows PowerShell cmdlets. For evaluation purposes, you should install a single runbook worker on the same computer as the web service.
- Windows Server 2012 R2 is supported in this release.

In addition, in System Center 2012 R2, Orchestrator has the following new and updated integration Packs (IPs):

New in System Center 2012 R2

System Center Integration Pack for Microsoft SharePoint

Updated in System Center 2012 R2

Windows Azure Integration Pack for Orchestrator in System Center 2012 SP1 and System Center 2012 R2

System Center Integration Pack for System Center 2012 Virtual Machine Manager

What's New in Orchestrator in System Center 2012 Service Pack 1

The following are the new features for Orchestrator in System Center 2012 Service Pack 1 (SP1).

What's New

In System Center 2012 Service Pack 1 (SP1), Orchestrator has the following new and updated integration Packs (IPs).

New in System Center 2012 SP1

Exchange Administrator Integration Pack for Orchestrator in System Center 2012 SP1
--

Exchange Users Integration Pack for Orchestrator in System Center 2012 SP1
--

Representational State Transfer (REST)
--

Integration Pack Guide for Orchestrator in System Center 2012 SP1

Updated in System Center 2012 SP1

Active Directory Integration Pack for System Center 2012 - Orchestrator

HP Service Manager Integration Pack for System Center 2012 - Orchestrator

System Center Integration Pack for System Center 2012 Operations Manager
--

System Center Integration Pack for System Center 2012 Virtual Machine Manager

VMware vSphere Integration Pack for System Center 2012 - Orchestrator

Orchestrator Capabilities

IT administrators perform many tasks and procedures to keep the health of their computing environment up-to-date and their business running. Tasks might include the following diverse activities, for example, new employees require that accounts and resources are configured, a business acquisition requires integrating a system from another vendor, and new hardware requires provisioning. Individual tasks and subtasks are automated, but typically, not the whole process. In addition, the administrators must maintain quality standards and system efficiency. System Center 2012 - Orchestrator can tie disparate tasks and procedures together by using the graphical user-interface Runbook Designer to create reliable, flexible, and efficient end-to-end solutions in the IT environment.

By using Orchestrator, you can carry out the following tasks:

- Automate processes in your data center, regardless of hardware or platform.
- Automate your IT operations and standardize best practices to improve operational efficiency.
- Connect different systems from different vendors without having to know how to use scripting and programming languages.

Custom automation

Orchestrator provides tools to build, test, debug, deploy, and manage automation in your environment. These automated procedures, called runbooks, can function independently or start other runbooks. The standard activities defined in every installation of Orchestrator provide a variety of monitors, tasks, and runbook controls with which you can integrate a wide range of system processes. Each activity in a runbook publishes data that is available to any subsequent

activity in that runbook. You use this Published Data to provide dynamic, decision-making capabilities, which can include creating emails, alerts, log files, accounts, and more.

Your IT organization can use Orchestrator to improve efficiency and reduce operational costs to support cross-departmental objectives. Orchestrator provides an environment with shared access to common data. By using Orchestrator, you can evolve and automate key processes between groups and consolidate repetitive manual tasks. You can automate cross-functional team processes and enforce best practices for incident, change, and service management by creating runbooks that are customized for your requirements. Through automation, regularly recurring tasks reduce the number of manual and error-prone activities in your environment. Orchestrator helps you improve the reliability and predictability of your IT procedures.

Cross-platform integration

Orchestrator integrates with System Center, other Microsoft products, and non-Microsoft products to enable interoperability across the data center. Orchestrator improves efficiency across multiple tools, systems, and departments by eliminating or crossing technology and organizational process structures. You can extend the capabilities of Orchestrator with integration packs that include additional functionality for both Microsoft and non-Microsoft products and technologies. Orchestrator activities and integration packs reduce unanticipated errors and shorten service delivery time by automating the common tasks associated with enterprise tools and products.

End-to-end orchestration

Orchestration is the collective name for the automated arrangement, coordination, and management of systems, software, and practices. It enables the management of complex cross-domain processes. Orchestrator provides the tools for orchestration to combine software, hardware, and manual processes into a seamless system. These tools let you connect and automate workflows.

Just as manufacturing companies have automated common and repeatable tasks from their production processes, you can adopt this same efficiency in the IT environment by using Orchestrator to seamlessly perform and monitor your IT processes. Orchestrator can handle routine tasks, process enforcement, and reliably meet the demands of the largest enterprises. Orchestrator integrates seamlessly with other System Center products to integrate IT administrative tasks from start to finish.

Extensible structure

If you have a custom in-house solution, Orchestrator provides extensible integration to any system through the Orchestrator Integration Toolkit. You can create custom integrations that allow Orchestrator to connect to any environment.

Orchestrator uses a Representational State Transfer (REST)-based web service that can perform processes like start and stop runbook jobs and get reporting information in Open Data protocol (OData) format. The web service lets you develop applications that can use live data from Orchestrator.

See Also

[Getting Started with System Center 2012 - Orchestrator](#)

Orchestrator Architecture

This topic provides an overview of System Center 2012 - Orchestrator, including a description of the system architecture, the internals of a typical runbook workflow, and the flow of a deployed runbook.

Orchestrator deployment basics

The following table lists the features in a basic deployment of Orchestrator.

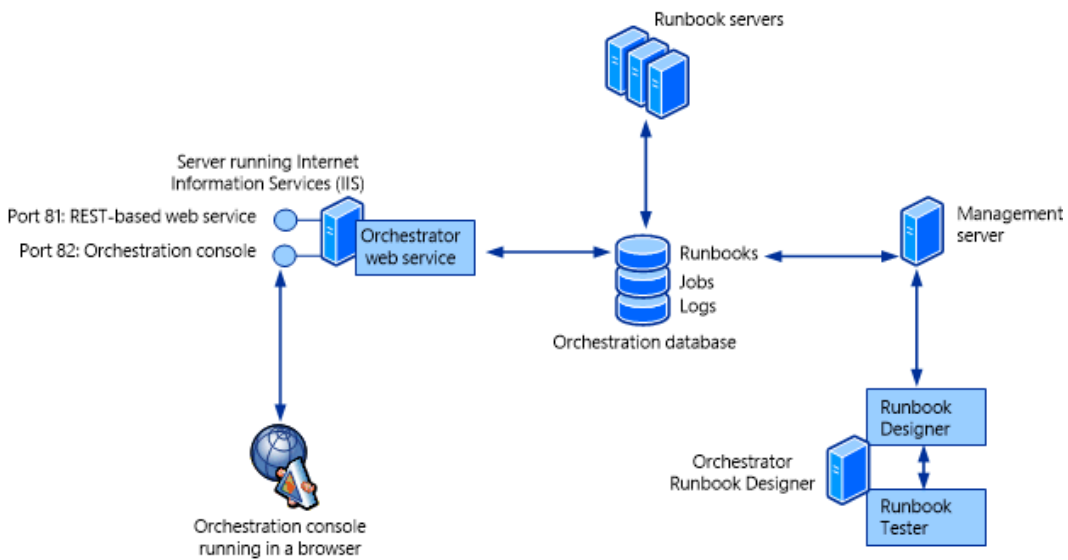
Orchestrator feature	Description
management server	The management server is the communication layer between the Runbook Designer and the orchestration database.
runbook server	A runbook server is where an instance of a runbook runs. Runbook servers communicate directly with the orchestration database. You can deploy multiple runbook servers per Orchestrator installation to increase capacity and redundancy.
orchestration database	The database is a Microsoft SQL Server database that contains all of the deployed runbooks, the status of running runbooks, log files, and configuration data for Orchestrator.
Runbook Designer	The Runbook Designer is the tool used to build, edit, and manage Orchestrator runbooks. For more information about the Runbook Designer, see System Center Orchestrator 2012 Runbook Guide .
Runbook Tester	Runbook Tester is a run-time tool used to test runbooks developed in the Runbook Designer. For more information about Runbook Tester, see How to Test a Runbook in System Center Orchestrator 2012 Runbook Guide .
Orchestration console	The Orchestration console lets you start or stop runbooks and view real-time status on a web browser. For more information about using the

Orchestrator feature	Description
	Orchestration console, see System Center Orchestrator 2012 Orchestration Console Guide .
Orchestrator web service	The Orchestrator web service is a Representational State Transfer (REST)-based service that enables custom applications to connect to Orchestrator to start and stop runbooks, and retrieve information about operations by using custom applications or scripts. The Orchestration console uses this web service to interact with Orchestrator.
Deployment Manager	Deployment Manager is a tool used to deploy integration packs (IPs), runbook servers, and Runbook Designers. For more information about this tool, see System Center Orchestrator 2012 Deployment Guide .

Architectural diagram

The following diagram illustrates each of the Orchestrator features and the communication between each.

System Center 2012 - Orchestrator Architecture



The orchestration database is the center of the Orchestrator installation containing all runbooks, configuration settings, and logs. The management server is required as a communication layer

between the Runbook Designer and the orchestration database. One or more runbook servers communicate directly with the database to retrieve runbooks to run and store information about the jobs created from the runbooks. The web service also communicates directly with the orchestration database and provides a web browser connection for the Orchestration console.

Orchestrator Extensions

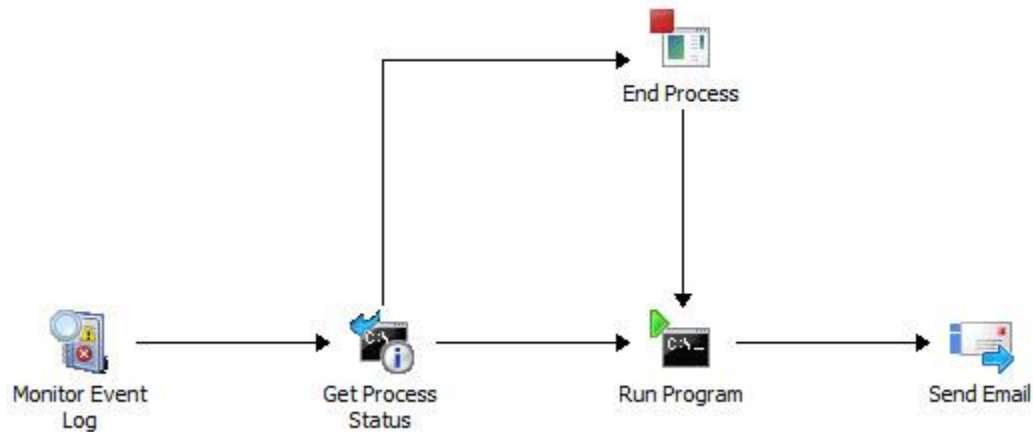
The following table shows multiple strategies available for extending the functionality provided by a standard installation of Orchestrator. For additional information, see **System Center Orchestrator 2012 Deployment Guide**.

Orchestrator feature	Description
integration pack (IP)	An integration pack is a collection of custom activities specific to a product or technology. Microsoft and other companies provide integration packs with activities to interact with their product from an Orchestrator runbook.
Orchestrator Integration Toolkit	The Orchestrator Integration Toolkit lets you extend your library of activities beyond the collection of standard activities and integration packs. The Integration Toolkit has wizard-based tools to create new activities and integration packs for Orchestrator. Developers can also use the Integration Toolkit to create integration packs from custom activities that they build by using the Orchestrator SDK.

Automation by using runbooks

To automate a task or process in Orchestrator, you use the Runbook Designer to create a runbook. You add activities to the runbook by dragging them from the **Activities** pane, and then link activities in the required order to create a workflow.

The following illustration shows a simple runbook.



This runbook monitors an event log. When it detects the specified event, the runbook checks the status of a particular process in Windows on a specific computer. If the process is found to be running, it is stopped. The runbook then starts the process and sends an email as a notification of the change of process state.

Each runbook activity finishes before proceeding to the next, and activities are available that provide complex logic such as requiring that multiple activities are completed before the runbook proceeds. By using a combination of logic on activities and smart links, you can implement whatever logic your particular automation scenario requires.

How Orchestrator processes a Runbook

After you have created a runbook, you commit it to the orchestration database by checking it in. You can then use either the Runbook Designer or the Orchestration console to start and stop the runbook.

A request to run a runbook creates a *job* that is stored in the orchestration database. Each runbook can define a primary runbook server and one or more standbys that process the runbook if the primary is unavailable. A service on each runbook server continuously monitors the orchestration database for jobs that it can process. When a runbook server detects a job, it logs that it is working on the job, copies the runbook locally, logs that it is running an instance of the runbook, and then begins processing the runbook. For any runbook not containing a monitor, you can create multiple runbook requests meaning that a single runbook can have multiple jobs.

When a runbook server processes a job, it creates an *instance* of the runbook by making a copy of it locally, and then performing the actions defined within the runbook according to the included workflow logic. Status information, activity results, and data are recorded in the orchestration database so that you can monitor the real-time and historical status of the runbook.

Permissions

Access to Orchestrator is provided by adding user accounts to a security group that is created during installation. This group can either be a domain group or a local group on the management server. Users of this group have full access to the Runbook Designer to create and modify runbooks and the Deployment Manager to deploy new Runbook Designers and runbook servers. Operators who have to start and stop runbooks but not create them can be granted this permission to individual runbooks and then use the Orchestration console.

Orchestrator Terminology

This topic provides terms and definition for System Center 2012 - Orchestrator and shows changes of Opalis Integration Server 6.3 terms to Orchestrator terminology and their definitions.

Terms and definitions

- [Glossary for System Center 2012 - Orchestrator](#)
Provides definitions for common terminology used in Orchestrator.
- [Glossary for Opalis Integration Server 6.3](#)
Provides definitions for common terminology used in Opalis Integration Server 6.3. Identifies changes between Opalis Integration Server 6.3 and Orchestrator.

See Also

[Getting Started with System Center 2012 - Orchestrator](#)

Glossary for System Center 2012 - Orchestrator

Term	Definition
activity	A single task in a runbook that performs a specific function.
check in	To save the changes in a runbook to the database.
check out	To allow edits to a runbook.
counter	A global integer variable that is used in a runbook.
data bus	A mechanism in Orchestrator that passes information from one activity in a runbook

Term	Definition
	to another activity.
instance	A unique occurrence of a runbook that is running on a runbook server.
integration pack	A collection of custom activities that is specific to a product or a technology.
IP	See Other Term: integration pack
job	A request to run a runbook.
junction	A runbook activity that synchronizes multiple branches of a runbook.
management server	The communication layer between the Runbook Designer and the deployment manager to the database.
monitor	An activity that continuously runs and that initiates a runbook when the monitor matches the criteria that you specify.
OIT	See Other Term: Orchestrator Integration Toolkit
Orchestration console	A web-based console that you can use to start, stop, and view information about runbooks.
orchestration database	The Oracle or SQL Server database where configuration information, runbooks, and logs are stored.
Orchestrator Integration Toolkit	A set of software tools that you can use to create custom integration packs.
Published Data	The data that is published to the databus from each activity in a runbook.
runbook	The sequence of activities that orchestrate actions on computers and networks.
Runbook Designer	The tool that is used by designers to create, modify, and deploy runbooks.
runbook server	The server that runs the service that manages runbooks and communicates with the orchestration database.
Runbook Tester	The tool that is used to test and validate

Term	Definition
	runbooks.
schedule	The global settings that you can use to define a set of date and time criteria for a runbook.
smart link	The connection between two activities in a runbook.
standard activity	The set of activities that is included with the standard installation of Orchestrator.
subscribe	To request data from the data bus.
variable	A global value that is used to define a frequently used setting, such as a directory path to common files or server names.

Glossary for Opalis Integration Server 6.3

The following table lists Opalis Integration Server 6.3 terms and the Orchestrator terms that replace them. A brief definition is included for each term.

Opalis Integration Server 6.3 term	System Center 2012 - Orchestrator term	Definition
Action server	runbook server	A runbook server is a computer that receives an instance of a runbook and runs the sequence of activities. Runbook servers communicate directly with the orchestration database; they do not require a management server to run runbooks.
Client	Runbook Designer	See definition for Opalis client.
custom start	initialize data	The initial runbook activity defined in a runbook to provide user-defined input parameters for the runbook.
datastore	orchestration database	The orchestration database is a SQL Server database

Opalis Integration Server 6.3 term	System Center 2012 - Orchestrator term	Definition
		containing configuration information, runbooks, and logs for Orchestrator.
foundation object	standard activity	The set of runbook activities available in a default installation. This includes monitors, tasks, and all runbook controls.
object	activity	The tasks used to create a runbook.
Object palette	Activities pane	The Activities pane is located in the tasks pane in the Runbook Designer. Collections of activities are grouped by function or integration pack.
Opalis client	Runbook Designer	An application used to create, modify, and deploy runbooks.
Operator console	Orchestration console	The interface that enables a user to see available runbooks, the real-time status of jobs and running instances, view their status, and start or stop runbooks, jobs, or instances.
Policy	runbook	A runbook is a collection of activities that orchestrates actions, events, and tasks.
Policy folder	runbook folder	A folder that contains one or more runbooks.
policy module	job process	A request to run a specific runbook that is waiting for assignment to a runbook server for processing.
Policy Testing Console	Runbook Tester	The tool used by Runbook Designers to test policies

Opalis Integration Server 6.3 term	System Center 2012 - Orchestrator term	Definition
		before deployment.
publish policy data	Published Data	Published Data is a runbook activity used to publish data from the runbook back to a calling (parent) runbook.
request	job	A job is a request to deploy and run a runbook on a runbook server. Jobs are stored in the orchestration database queue.
trigger policy	Invoke Runbook	An Invoke Runbook activity calls another runbook from within a runbook. The Invoke Runbook activity can optionally wait for the called runbook to finish before proceeding. Data is returned from the invoked runbook by using the Returned Data activity. It is equivalent to the function call found in many programming languages.
workflow control	runbook control	A collection of standard activities that manage how runbook logic behaves.

Orchestrator Resources

In addition to this online reference for System Center 2012 - Orchestrator, there are a number of resources that can provide additional information about building runbooks, by using [System Center 2012 - Orchestrator SDK](#) and applying best practices.

Resource	Location
System Center 2012 - Orchestrator Home	http://www.microsoft.com/systemcenter/orchestrator

Resource	Location
System Center Home on TechNet	http://technet.microsoft.com/systemcenter/
Orchestrator Team Blog on TechNet	http://blogs.technet.com/b/orchestrator/
Orchestrator Community Releases on CodePlex	http://orchestrator.codeplex.com
Orchestrator Community Forums on TechNet	http://social.technet.microsoft.com/Forums/category/systemcenterorchestrator

See Also

[Getting Started with System Center 2012 - Orchestrator](#)