System Center Orchestrator 2012 Getting Started Guide

Microsoft Corporation

Published: September 2011

Feedback

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

Copyright

Contents

System Center Orchestrator 2012 Getting Started Guide

Orchestrator is a workflow management solution for the datacenter. It enables you to automate the creation, monitoring, and deployment of resources in your environment. The information contained in this document describes the architecture of Orchestrator. The guide includes terminology changes from Opalis 6.3, definitions of key terms and concepts, and information on where to find additional resources. After reading this guide, you should have a basic understanding of how Orchestrator works and where you can go for more information.

Getting Started Topics

This document covers the following topics:

Other resources for this product

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

Orchestrator Capabilities

IT administrators perform many tasks and procedures to keep their computing environment health and up to date and to keep their business running. New employees need accounts and resources configured; a business acquisition requires integrating a system from another vendor; new hardware needs provisioning. Individual tasks and subtasks are automated, but typically, not the entire process. Maintaining quality standards and system efficiency is difficult. Orchestrator can tie disparate tasks and procedures together using a visual Runbook Designer to create reliable, flexible, and efficient end-to-end solutions in your IT environment.

System Center Orchestrator 2012 enables you to automate processes in your data center, regardless of hardware or platform. Using Orchestrator, you can automate your IT operations and standardize on best practices to improve operational efficiency. Orchestrator allows you to

connect different systems from different vendors without any knowledge of scripting or 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 launch other runbooks. The standard activities defined in every installation of Orchestrator provide a variety of monitors, tasks, and runbook controls that allow you to 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 the creation of emails, alerts, log files, accounts, and more.

Your IT organization can use Orchestrator to improve efficiency and reduce operational costs to support cross-department objectives. Orchestrator provides an environment with shared access to common data. Using Orchestrator you can evolve and automate key processes between groups, reduce repetitive manual tasks, and obtain the highest value from your skilled workers. Automate cross-silo processes and enforce best practices for incident, change, and service management by creating runbooks customized for your needs. Through automation, regularly recurring tasks reduce the number of manual and error-prone activities in your environment. Orchestrator helps you to 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 datacenter. It improves efficiency across multiple tools, systems, and departments by eliminating or crossing technology and organizational process silos. 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 automated arrangement, coordination, and management of systems, software, and practices. It enables the management of complex cross-domain processes. Orchestrator provides the tools to combine software, hardware, and manual processes into a seamless system, allowing you to connect and automate workflows.

Manufacturing companies have automated common and repeatable tasks from their production processes. You can adopt this same efficiency in your IT environment 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 beginning to end.

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 utilizes a REST-based web service that provides the ability to do things like start and stop runbook jobs as well as get information in OData format, allowing you to develop applications that can use live data from Orchestrator.

Orchestrator Architecture

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

Orchestrator Deployment Basics

A basic deployment of Orchestrator consists of the following features:

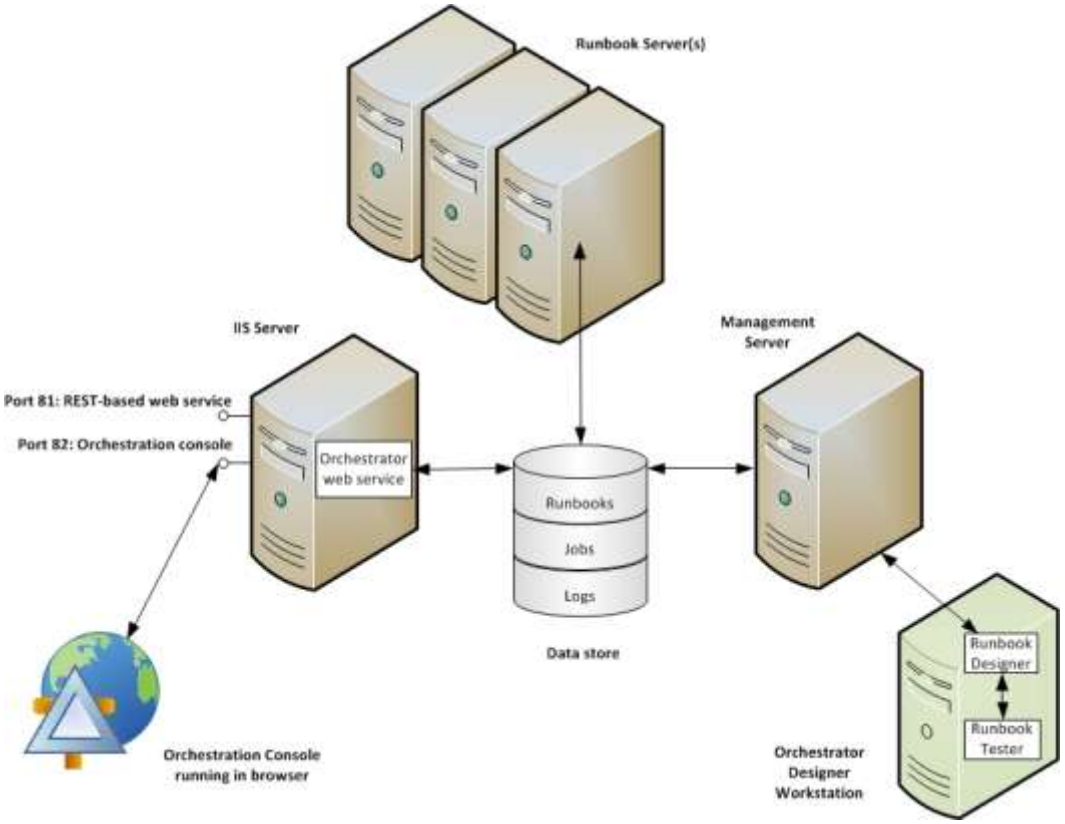
Orchestrator Feature	Description
Management server	The Management server is the communication layer between the Runbook Designer and the database.
Runbook Server	A Runbook Server is where an instance of a runbook runs. Runbook Servers communicate directly with the Orchestrator database. You can deploy multiple runbook servers per Orchestrator installation for capacity and redundancy.
Orchestrator database	The database is a Microsoft SQL Server database that contains all of the deployed runbooks, 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 on the Runbook Designer, see **System Center Orchestrator 2012 Runbook Guide**.
Runbook Tester	The Runbook Tester is a runtime tool used to test runbooks developed in the Runbook Designer. For more information on the

Orchestrator Feature	Description
	Runbook Tester, see **How to Test a Runbook** in **System Center Orchestrator 2012 Runbook Guide**.
Orchestration console	The Orchestration console enables you to start or stop runbooks and view real-time status from a web browser. For more information on using the Orchestration console, see the **System Center Orchestrator 2012 Orchestration Console Guide**
Orchestrator web service	The Orchestrator web service is a REST-based service that enables custom applications to connect to Orchestrator to start and stop runbooks, and retrieve information about operations using custom applications or scripts. The Orchestration console uses this web service to interact with Orchestrator.
Deployment Manager	The Deployment manager is a tool used to deploy integration packs (IPs), Runbook servers, and Runbook Designers. For more information on this tool, see the **System Center Orchestrator 2012 Deployment Guide**.

Architectural diagram

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

Orchestrator Architecture



The 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 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 database and provides a connection from client browsers for the Orchestration console.

Orchestrator Extensions

There are multiple strategies available for extending the functionality provided by a standard installation of Orchestrator as shown the following table. . See the **System Center Orchestrator 2012 Deployment Guide** for additional information.

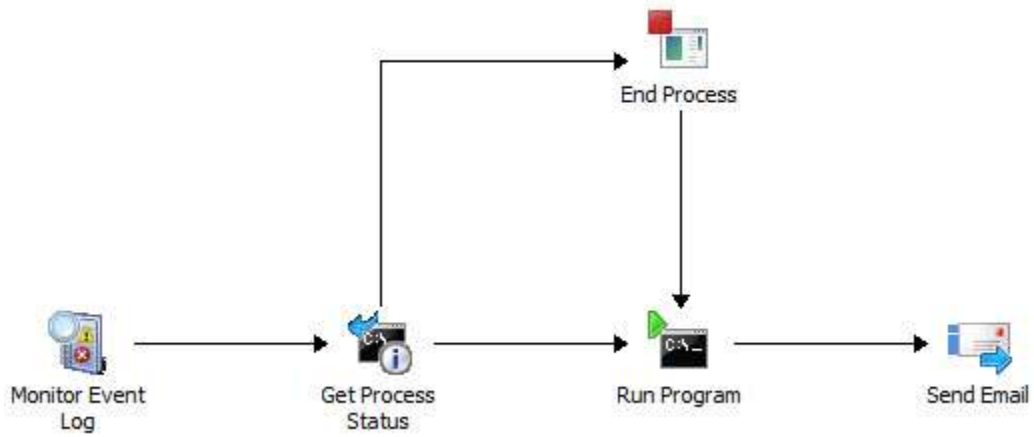
Orchestrator Feature	Description
integration pack (IP)	An integration pack is a collection of custom activities specific to a product or technology. Microsoft and third parties provide integration packs with activities to interact with their product from an Orchestrator runbook.

Orchestrator Feature	Description
Orchestrator Integration Toolkit	The Orchestrator Integration Toolkit allows you to extend your library of activities beyond the collection of standard activities and integration packs. The Integration Toolkit has wizard-based utilities to create new activities and integration packs for Orchestrator. Developers can also use the Integration Toolkit to create integration packs from custom activities built using the Orchestrator SDK.

Automation 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 Activity pane, and then link activities in the required order to create a workflow.

The illustration below shows a simple runbook.



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

Each runbook activity completes 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. 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

Once you have created a runbook, you commit it to the Orchestrator 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 database. Each runbook can define a primary runbook server and one or more standbys that will process the runbook if the primary is unavailable. A service on each Runbook server continuously monitors the database for jobs that it can process. Once 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, then begins processing the runbook. For any runbook not containing a monitor, you can create multiple start 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 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 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 need 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

System Center Orchestrator 2012 uses a number of terms and concepts defined below.

Terms and Definitions

The following table contains common Orchestrator terms and their definitions.

Orchestrator Term	Definition
activity	A type of action in a runbook that performs a specific function. You create a runbook by joining together multiple activities that run in a defined sequence
activity instance	An execution of an activity within a runbook instance
check in / check out	To modify a runbook, the runbook must be

Orchestrator Term	Definition
	checked out. A runbook cannot be started while checked out. After updating a runbook, checking it in commits changes to the database.
counter	A global integer variable used in a runbook. The counter is modifiable within a runbook, such as counting an incremental number of attempts of an action.
data bus	A mechanism used within Orchestrator to pass information from one activity in a runbook to the next activity. The data output from each activity to the data bus is called *published data*. Each activity has access to the data published by all previous activities in the runbook.
database	The database is the SQL Server database where configuration information, runbooks, and logs are stored.
instance	An instance is a unique occurrence of a runbook, running on a Runbook server.
integration pack (IP)	A collection of custom activities specific to a product or a technology, contained in a package deployable by the Deployment Manager.
junction	A runbook activity that synchronizes multiple branches of a runbook.
job	A request to run a runbook
Management server	The communication layer between the Runbook Designer and the Deployment Manager to the database.
monitor	A specific activity type often used to initiate a runbook. A monitor runs continuously and invokes an action when specified criteria is met.
Orchestration console	A web-based console that allows you to start, stop, and view information on runbooks.
published data	The data resulting from each activity in a runbook. This data is published to the data bus.

Orchestrator Term	Definition
	Activities in the runbook can subscribe to the published data values to use as input property values. Smart link conditions also use this information to add decision-making to the workflow.
runbook	A defined set of actions to perform. Runbooks contain one starting point, but can contain branches, actions, and multiple stopping points.
Runbook Designer	The tool that designers use to create, modify, and publish runbooks.
Runbook server	The server that runs the activities defined in a runbook. Runbook servers communicate directly with the database. You can deploy one or more Runbook Servers in an Orchestrator installation for capacity and redundancy.
Runbook Tester	The tool used to test and validate runbooks.
schedule	Global settings that allow you to define a set of date/time criteria for a runbook.
smart link	The connection between two activities in a runbook. Smart links provide conditional flow from one activity to another based on criteria defined in the filters.
standard activities	The set of activities included with the standard installation of Orchestrator.
variable	A global value used in a runbook to define a frequently used setting, such as a directory path to common files, server names, or other strings.

Opalis Integration Server 6.3 Terminology

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 Orchestrator 2012 term	Definition
Action server	Runbook server	A Runbook Server is a computer that receives an instance of a runbook and executes the sequence of activities. Runbook Servers communicate directly with the database; they do not require a Management Server 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	database	The database is a Microsoft SQL Server database 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 on the right-hand side 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

Opalis Integration Server 6.3 term	System Center Orchestrator 2012 term	Definition
		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 before deployment.
publish policy data	Returned Data	Returned 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 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 complete before proceeding. Data is returned from the invoked runbook using the Return 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 provided for System Center Orchestrator 2012, there are a number of resources that can provide additional information on building runbooks, using the Orchestrator Integration Toolkit, and best practices.

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