Kerrie Meyler
Pete Zerger
Marcus Oh
Kurt Van Hoecke
Anders Bengtsson

MVP
Microsoft®
Most Valuable
Professional

# System Center Orchestrator 2012

## UNLEASHED

SAMS

Kerrie Meyler
Pete Zerger
Marcus Oh
Anders Bengtsson
Kurt Van Hoecke

with Romuald Gauvin
Nicholas J. Dattilo

# System Center 2012 Orchestrator

## UNLEASHED

SAMS | 800 East 96th Street, Indianapolis, Indiana 46240 USA

**System Center 2012 Orchestrator Unleashed**

ISBN-13: 978-0-672-33610-2

ISBN-10: 0-672-33610-3

Library of Congress Control Number: 2013943417

Printed in the United States of America

First Printing September 2013

## Trademarks

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Sams Publishing cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

## Warning and Disclaimer

Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied. The information provided is on an "as is" basis. The authors and the publisher shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the CD or programs accompanying it.

## Bulk Sales

Sams Publishing offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales. For more information, please contact

**U.S. Corporate and Government Sales**
**1-800-382-3419**
**corpsales@pearsontechgroup.com**

For sales outside of the U.S., please contact

**International Sales**
**international@pearsoned.com**

**Editor-in-Chief**
Greg Wiegand

**Executive Editor**
Neil Rowe

**Development Editor**
Mark Renfrow

**Managing Editor**
Kristy Hart

**Senior Project Editor**
Lori Lyons

**Project Editor**
Katie Matejka

**Copy Editor**
Krista Hansing

**Indexer**
Erika Millen

**Proofreader**
Sarah Kearns

**Technical Editor**
Jeff Fanjoy

**Editorial Assistant**
Cindy Teeters

**Cover Compositor**
Mark Shirar

**Senior Compositor**
Gloria Schurick

# Contents at a Glance

# Table of Contents

# Foreword

*I have seen automation defined as the use of machines, controls, and information technologies to optimize the productivity in the production of goods and delivery of services.*

In today's modern data centers, this statement is both true and a requirement that allows your IT people the assets to work on strategic initiatives and spend less time on repetitive, mundane tasks that can be susceptible to human error. On the other hand, automation will not achieve these gains in and of itself, as data centers across the globe are built, managed, and sustained using a multitude of workloads that provide a service to users and customers. Enter *integration*—when automation and integration intersect, IT departments are provided the tools necessary to reach into disparate systems and essentially get them to "talk" to one another using well-defined workflows or *runbooks* as we sometimes call them. These runbooks allow IT staff to compose highly available, flexible automation and integration touch-points across business processes that span a multitude of workloads on various platforms.

Designing, publishing, and executing these workflows is simple using System Center 2012 SP 1 Orchestrator, a System Center 2012 SP 1 component. Orchestrator allows IT staff to deploy *integration packs* for all the System Center components as well as third-party workloads such as HP, IBM, VMware, and also other Microsoft workloads outside System Center such as Active Directory, Exchange, FTP, REST, and Windows Azure. Composing these workflows is easy using the Orchestrator Runbook Designer, which provides the user with a simple WYSIWYG graphical interface for dragging and dropping activities into a sequence that makes sense to your defined business process. Once enabled, these workflows can be manually triggered from the designer or invoked from another system such as System Center Service Manager. Users also have the ability to execute these workflows from our RESTful web service without requiring the Runbook Designer. Combine this designer with a highly available SQL Server backend and runbook servers that are able to scale out, and you have an enterprise-ready automation and integration tool that is simple to use and powerful enough to automate away business processes within your organization.

The demand for automation and integration has been quickly trending upward in IT. No matter whom you talk to, from large to small, automating business processes is becoming more prevalent in organizations around the world. This book intends to instruct IT administrators on how to use System Center 2012 Orchestrator to integrate and automate their existing business processes using a friendly, easy-to-use WYSIWYG designer with ready-to-import integration packs that cover a multitude of workloads essential to your business. The authors asked me to provide the Foreword for the book; and I can't think of a better-suited group of individuals who are able to produce this type of documentation, examples, and real-world scenarios to help you take advantage of this powerful System Center 2012 component.

**Justin Incarnato, Senior Program Manager**
**Cloud and Enterprise Division, Microsoft**

# About the Authors

**Kerrie Meyler**, System Center MVP for Cloud and Datacenter Management, is the lead author of numerous System Center books in the Unleashed series. This includes *System Center Operations Manager 2007 Unleashed* (2008), *System Center Configuration Manager 2007 Unleashed* (2009), *System Center Operations Manager 2007 R2 Unleashed* (2010), *System Center Opalis Integration Server 6.3 Unleashed* (2011), *System Center Service Manager 2010 Unleashed* (2011), *System Center 2012 Configuration Manager Unleashed* (2012), and *System Center 2012 Operations Manager Unleashed* (2013). She is an independent consultant and trainer with more than 15 years of Information Technology experience. Kerrie has presented on System Center technologies at TechEd NA and MMS.

**Pete Zerger** is a consultant, author, speaker, and System Center Cloud and Datacenter Management MVP focusing on System Center management, private cloud, and data center automation solutions. He is a frequent speaker at Microsoft conferences, and writes articles for a variety of technical magazines including *Microsoft TechNet*. Pete is a contributing author for several books, including *System Center Opalis Integration Server 6.3 Unleashed* (2011), *PowerShell 2.0 Bible* (Wiley, 2011), and *System Center 2012 Operations Manager Unleashed* (2013). He is also the co-founder of SystemCenterCentral.com, a popular web community providing information, news, and support for System Center technologies. In 2008, Pete founded the System Center Virtual User Group, a group dedicated to sharing System Center knowledge with users worldwide.

**Marcus Oh**, System Center Cloud and Datacenter Management MVP, is a senior technical manager for a large telecommunications provider, running directory services and management infrastructure for ~30,000 systems. He has been an MVP since 2004 in System Center, specializing in Configuration Manager, Operations Manager, and Orchestrator. Marcus has written numerous articles for technology websites and blogs on Orchestrator and other System Center components at http://marcusoh.blogspot.com. He coauthored *Professional SMS 2003, MOM 2005, and WSUS* (Wrox, 2006), was a contributing author to *System Center Opalis Integration Server 6.3 Unleashed* (2011), and coauthored *System Center 2012 Configuration Manager Unleashed* (2012). Marcus is also the president of the Atlanta Systems Management User Group (http://www.atlsmug.com) and a board member of the Deskside Management Forum.

**Anders Bengtsson** is a Microsoft senior premier field engineer focusing on System Center. He has written a number of System Center training courses, including the Service Manager 2010 and Operations Manager 2007 advanced courses for Microsoft Learning. He was a coauthor for *System Center Service Manager 2010 Unleashed* (2011). Before joining Microsoft, Anders was a Microsoft MVP from 2007-2010 for his work in the System Center community, including more than 10,000 posts in news groups and forums. Anders has presented and worked at numerous Microsoft conferences and events, including MMS and Microsoft TechEd NA and EMEA.

**Kurt Van Hoecke**, System Center Cloud and Datacenter Management MVP, is a managing consultant at Inovativ Belgium. He focuses on the System Center, including Orchestrator, Service Manager, and Configuration Manager. Kurt was a contributing author to *System Center Service Manager 2010 Unleashed* (2011) and blogs for System Center User Group Belgium and AuthoringFriday, where he shares his field experiences and discusses how to extend the built-in functionality of the System Center components.

# About the Contributors

**Romuald Gauvin**, System Center Cloud and Datacenter Management MVP, manages a consulting company dedicated to orchestration and cloud projects. Previously in charge of development at Opalis Inc., Romuald began consulting in 2000. He has worked on automation and orchestration projects using Opalis/Orchestrator for more than 15 years. Romuald regularly shares his experiences on orchestration project approaches, methodology, and technical aspects during seminars with Microsoft, the French System Center user group, and at Microsoft TechDays.

**Nicholas J. Dattilo** is a consultant with Acceleres, a Microsoft Silver Partner for Management and Virtualization. He has worked with a wide range of clients to help them implement and optimize System Center Orchestrator and Service Manager. Nick frequently contributes to the monthly Acceleres Presents! webcasts on service delivery and automation.

# Dedication

*To those IT professionals worldwide interested in automation and
using System Center, and the System Center Cloud and
Data Center Management MVPs.*

# Acknowledgments

Writing a book is an all-encompassing and time-consuming project, and this book certainly meets that description. The authors and contributors would like to offer appreciation to those who helped with *System Center 2012 Orchestrator Unleashed*.

Thank you to Didier Leclercq of Aezan, to ClearPointe Technology for lab assistance and to John Joyner for environment support, and to Justin Incarnato of Microsoft. Jeff Fanjoy, also of Microsoft, was invaluable as our technical editor.

Thanks also go to the staff at Pearson, in particular to Neil Rowe, who has worked with us since with *Microsoft Operations Manger 2005 Unleashed* (Sams, 2006).

# We Want to Hear from You!

As the reader of this book, *you* are our most important critic and commentator. We value your opinion and want to know what we're doing right, what we could do better, what areas you'd like to see us publish in, and any other words of wisdom you're willing to pass our way.

We welcome your comments. You can email or write to let us know what you did or didn't like about this book—as well as what we can do to make our books better.

*Please note that we cannot help you with technical problems related to the topic of this book.*

When you write, please be sure to include this book's title and author as well as your name and email address. We will carefully review your comments and share them with the author and editors who worked on the book.

**Email:** consumer@samspublishing.com

**Mail:** Sams Publishing
ATTN: Reader Feedback
800 East 96th Street
Indianapolis, IN 46240 USA

# Reader Services

Visit our website and register this book at informit.com/register for convenient access to any updates, downloads, or errata that might be available for this book.

*This page intentionally left blank*

# Introduction

In December 2009, Opalis Software, Inc. became a wholly owned subsidiary of Microsoft Corporation. Opalis, a leader in information technology process automation (ITPA) and run book automation (RBA), was best known for its Opalis Integration Server (OIS) software. As Brad Anderson said at the time, the acquisition was a pivotal piece for delivering on Microsoft's dynamic data center initiative, as it brought together Opalis Software's deep data center automation expertise with the integrated physical and virtualized data center management capabilities provided by Microsoft System Center (http://blogs.technet.com/b/systemcenter/archive/2009/12/11/microsoft-acquires-opalis-software.aspx).

As part of the acquisition, Opalis Software released OIS 6.2.2, a remediated version of 6.2.1. In November 2010, Microsoft released OIS 6.3, which became the "last OIS." Microsoft then further integrated OIS into System Center 2012 and rebranded it as System Center Orchestrator. Orchestrator enables Microsoft to integrate process automation into its vision of the data center.

ITPA is a powerful capability that can assist in streamlining IT operations by removing much of the overhead associated with manual responses to IT problems, whereas BPA concentrates on automating processes linked to the core business of an enterprise; these are often linked to data management. System Center Orchestrator, which incorporates an easy-to-use, drag-and-drop user interface, enables you to capture and document processes that encompass an entire IT organization. This is a core building block for the future of IT and is the foundation for the automation necessary to deliver cloud computing—self-adjusting tools of computing resources that can be tuned based on real-time events.

This book is divided into four sections:

Part I, "Orchestrator Overview and Concepts," includes an introduction to Orchestrator and discusses its history, internals, architectural concepts, and design concepts. These topics are discussed in Chapter 1, "Orchestration, Integration, and Automation," Chapter 2, "What's New in System Center 2012 Orchestrator," Chapter 3, "Looking Inside System Center 2012 Orchestrator," and Chapter 4, "Architectural Design."

Part II, "Installation and Implementation," steps through the installation process and discusses implementing Orchestrator:

- Chapter 5, “Installing System Center 2012 Orchestrator,” covers installation and OIS 6.3 migration.
- Chapter 6, “Using System Center 2012 Orchestrator,” provides an overview of how to use this System Center component.
- Chapter 7, “Runbook Basics,” covers the anatomy of a runbook and introduces the different types of activities included with Orchestrator 2012.
- Chapter 8, “Advanced Runbook Concepts,” goes deeper into runbook concepts, including scheduling, invoking child runbooks, looping, junctions, working with data, error handling, computer groups, variables, and counters.
- Chapter 9, “Standard Activities,” provides additional depth on the Orchestrator standard activities.
- Chapter 10, “Runbook and Configuration Best Practices,” covers best practices for runbooks and configuration.
- Chapter 11, “Security and Administration,” discusses the Orchestrator security model, and user roles and security.

Part III, “Integration Packs and the OIT,” focuses on integrating System Center Orchestrator into the data center through integration packs. IPs are software components that plug into the larger Orchestrator framework, and are designed around a series of atomic tasks targeted to a specific application. Orchestrator IPs are discussed in Chapter 12, “Orchestrator Integration Packs.” The System Center IPs are discussed in greater depth in the following chapters:

- Chapter 13, “Integration with System Center Operations Manager”
- Chapter 14, “Integration with System Center Service Manager”
- Chapter 15, “Integration with System Center Configuration Manager”
- Chapter 16, “Integration with System Center Virtual Machine Manager”
- Chapter 17, “Integration with System Center Data Protection Manager”

Chapter 18, “Integration with Windows Azure,” goes into depth on the Windows Azure IP, introduced with System Center 2012 Service Pack 1.

Chapter 19, “Runbook Automation in the Data Center and the Cloud,” takes the Azure and System Center IPs to the next level by presenting examples that integrate objects from these IPs together in workflows and also incorporate PowerShell to achieve true end-to-end automation. Just in case you still don’t have all the objects you need to accomplish your own integrations, Chapter 20, “The Orchestrator Integration Toolkit,” gives you the tools to create your own IPs using the Orchestrator Integration Toolkit, also known as the OIT.

By this time, you should have all the tools necessary to become an Orchestrator expert. The last section of the book includes three appendices. Appendix A, “Community

Solutions and Tools," includes resources developed by the community, Appendix B, "Reference URLs," incorporates useful references you can use for further information, and Appendix C, "Available Online," is a guide to supplementary resources offered with the book that you can download from Pearson's website at http://www.informit.com/store/product.aspx?isbn=9780672336102.

This book provides in-depth reference and technical information about System Center 2012 Orchestrator SP 1, as well as information on orchestrating with System Center and third-party products through integration packs. The material will be of interest to those shops using System Center, Orchestrator, and anyone interested in ITPA.

Microsoft announced System Center 2012 R2 at TechEd in early June 2013. This release, slated for general availability by the end of the year, provides parity between Microsoft's data center software and its public cloud portfolio. As such, there are minimal changes planned to Orchestrator 2012 beyond updates to the Azure and VMM IPs, a new IP for SharePoint, support for Windows Server 2012 R2, and updates to the Orchestrator installation program for installing the new Service Management Automation (SMA) web service and runbook workers. The SMA feature is also interesting in that it provides a glimpse to where Microsoft may go with cloud-based automation.

## Disclaimers and Fine Print

There are several disclaimers. Microsoft is continually improving and enhancing its products. This means the information provided is probably outdated the moment the book goes to print.

In addition, the moment Microsoft considers code development on any product complete, they begin working on a cumulative update, service pack, or future release; as the authors continue to work with the product, it is likely yet another one or two wrinkles will be discovered! The authors and contributors of *System Center 2012 Orchestrator Unleashed* have made every attempt to present information that is accurate and current as known at the time. Updates and corrections will be provided as errata on the InformIT website at http://www.informit.com/store/system-center-2012-orchestrator-unleashed-9780672336102.

Thank you for purchasing *System Center 2012 Orchestrator Unleashed*. The authors hope it is worth your while!

*This page intentionally left blank*

# PART I

# Orchestrator Overview and Concepts

## IN THIS PART

*This page intentionally left blank*

# CHAPTER 1

# Orchestration, Integration, and Automation

IN THIS CHAPTER

- ▶ Orchestration, ITIL, and MOF
- ▶ Integration, Automation, Orchestration: The Differences
- ▶ Positioning of Orchestrator in System Center 2012
- ▶ Typical Use Cases
- ▶ Managing a Project
- ▶ Example: Orchestrating a VM Deployment

Information technology's (IT) expansion in recent decades has enabled companies to define business-driven applications that can help improve their performance. With IT services now regarded as key to the growth of organizations, the health and availability of these services is critical. This realization has led to routine procedures to check system availability, determine whether these systems are running correctly, verify whether data processing occurred as expected last night, and so on. These procedures and checks to ensure availability and health become increasingly complex as IT's role in the business continues to grow.

As part of this growth in the role of IT and the expansion of distributed computing in the mid-1990s, Opalis Robot, the grandfather of System Center Orchestrator, came into existence. (See Chapter 2, "What's New in System Center 2012 Orchestrator," for a detailed history of Orchestrator.) The goal of Robot was to help IT teams manage routine procedures more efficiently and be able to react quickly to events (unexpected behaviors). Opalis Robot was an implementation of run book automation (RBA) software, which helps organizations manage runbooks of routine procedures.

The need for IT management activities remains unchanged today: IT must focus on delivering services to companies and ensuring that those services are available. These activities include tools and management processes such as the IT Information Library (ITIL). As IT continues to evolve, these are evolving as well.

Automation, the focus of System Center Orchestrator, includes additional approaches such as business process automation (BPA) and IT process automation (ITPA):

- BPA concentrates on automating processes linked to the core business of an enterprise and often processes linked to data management. As an example, a process could perform a daily extraction of new customers registered on a website, convert those into a CSV file, send the file by email to a person in charge of data quality to check validity, and then import the file into customer relationship management (CRM) software.
- ITPA focuses on orchestrating and integrating tools, people, and processes through automated workflows dedicated to IT activities. Those activities can be all types of routine IT procedures, such as deployment (virtual machine creation, application deployment, and so on) and configuration updates.

This chapter discusses integration, automation, and orchestration from an IT management point of view. It shows how System Center and Orchestrator can help you extend best practices of ITIL and the Microsoft Operations Framework (MOF), and how Orchestrator can help you integrate, automate, and orchestrate your IT environment.

## Orchestration, ITIL, and MOF

If you have already started an ITIL project in your organization, you have probably spent a considerable amount of time defining processes, process owners, and change policies. Improving quality in your organization can be a challenging task and requires an initial investment in time and resources.

ITIL is widely accepted as an international standard of best practices for operations management. Those ITIL projects are excellent opportunities to gain benefits from orchestration. As you inventory your processes, you can identify technical operations that appear regularly. For example, orchestration perfectly fits a scenario with multiple individuals collaborating to treat a user request. This could be delivering a new server or application or updating a user quota.

MOF is closely related to ITIL, and both describe best practices for IT service management (ITSM) processes. ITIL is generally accepted as the “best practices” for the industry. Because it is technology agnostic, it is a foundation that can be adopted and adapted to meet the specific needs of various IT organizations. Although Microsoft chose to adopt ITIL as a standard for its own IT operations for its descriptive guidance, the company designed MOF to provide prescriptive guidance for effective design, implementation, and support of Microsoft technologies. MOF is a set of publications that provide both descriptive (what to do, when, and why) and prescriptive (how to do) guidance on ITSM. The key focus in developing MOF was providing a framework specifically geared toward managing Microsoft technologies.

MOF v4 now incorporates Microsoft's previously existing Microsoft Solutions Framework (MSF), providing guidance for application development solutions. The combined framework provides guidance throughout the IT life cycle, as Figure 1.1 shows.

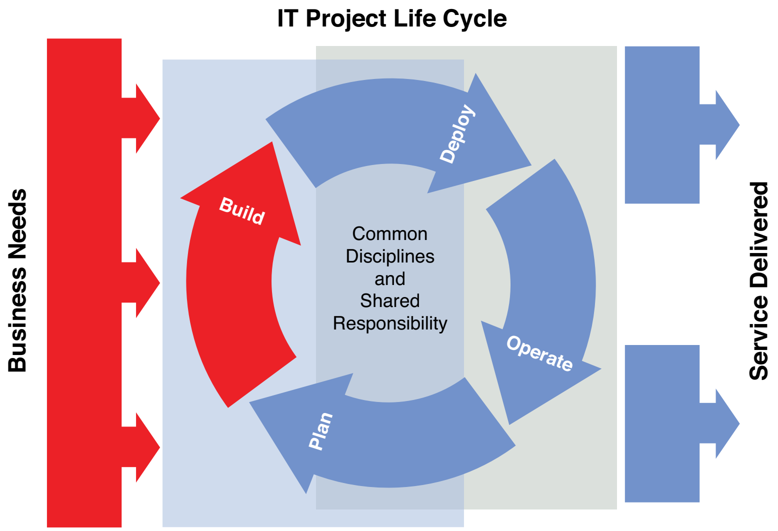


FIGURE 1.1 The IT lifecycle.

At its core, MOF is a collection of best practices, principles, and models. It provides direction on achieving reliability, availability, supportability, and manageability of mission-critical production systems by focusing on solutions and services that use Microsoft products and technologies. MOF extends ITIL by including guidance and best practices derived from the experience of Microsoft's internal operations groups, partners, and customers worldwide. MOF aligns with and builds on the ITSM practices documented within ITIL, thus enhancing the supportability built on Microsoft's products and technologies.

The authors recommend taking a new look at your ITIL processes to identify potentials for orchestration. You will gain valuable returns while ensuring that all processes are realized according to your ITIL definition—and you will save time on orchestrated actions.

## Integration, Automation, Orchestration: The Differences

As an IT Pro, you might at some point face situations in which you need to do the following:

- Regularly check the availability of an application, server, or process
- Set up new servers, applications, file shares, and services

- React to an event in the system
- Verify that applications have run correctly, but lack the time to do so
- Respond to a call from a user on something that is not working as expected

This list just touches the surface of what regularly occurs in IT. This is just one place where integration, automation, and orchestration can help. If you have been exposed to marketing material on Orchestrator or its predecessor, Opalis Integration Server (OIS), you have probably heard these terms more than once. The next sections discuss these concepts and the differences in their approaches.

## About Integration

The first step in saving time and ensuring efficiency, and also popularized by UNIX administrators, is using scripts. Instead of having to perform a sequence of tasks manually, you can use scripts to include those sequences in a single operation. You then launch a single command and check just one result—you can even ask your developers to create a special program to assist with this. A script also provides an audit trail of exactly what occurred so that you do not have to manually type commands or perform console clicks.

Here's an example of when incorporating a script is beneficial: launching a backup from the command line on different servers.

- You first write a script that takes the targeted server as a parameter and launches the backup with the appropriate parameters.
- You then write another script to launch this backup on all your servers.
- You can even script your processes so that applications stop before the backup and restart when the backup is complete, all backup files are moved to a remote location, monitoring stops and then restarts during the operation, and so on. Here you are integrating an entire process—involving different servers and applications—in a single command.

The popularity of scripting languages, particularly PowerShell, shows how efficient scripting is. Scripting is easy to access, requires few skills to get started, and is quick to implement. Not surprisingly, IT teams develop scripts dedicated to their activities at so-called silos, organized around server management, virtualization, network, database, and so on. These are known as inner-silo integration scripts. However, anyone who writes or uses scripts also knows that scripts bring their own challenges:

- Scripts are usually written for a specific purpose, so they do not take advantage of sharing and generalization.
- Scripts are frequently not documented or are under documented, making them difficult to maintain.

- Various people can write and exploit scripting in different ways and at different times, without centralized coordination or reporting required.
- Error checking is poorly implemented—errors typically are checked during execution, and manual corrections are made as necessary.
- Scripts frequently must be rewritten when applications change or new ones need to be added to the sequence.
- As processes evolve, script complexity grows and can become difficult to manage.

Although scripts are useful for starting integration and saving time on simple processes, they are difficult to manage over time and across teams. You need to connect these scripts across teams and silos.

As applied to System Center Orchestrator, integration is the connection created between Orchestrator and an integration service on the application or applications to be automated and orchestrated.

## Using Automation

After you create task sequences (scripts or programs) to begin the integration process, automation is the second step. Automation enables these task sequences to be launched at the proper moment. Batch schedulers (known as CronTab in the UNIX world) typically handle this work. For every device you manage and system you maintain, there seem to be at least that many other solutions available to do the job. Keeping up with all this without automation can be challenging.

Schedulers enable you to launch a backup script at the proper time. Using the example of the backup task in the previous section, the batch scheduler lets you initiate a backup every Saturday night. Those batch schedulers are useful for daily routine procedures and give you flexibility in several areas:

- Launching your scripts at the correct time
- Launching scripts in response to different events, such as file creation/modification/deletion, process start/end, a new entry in the event log, and so on
- Launching one script when another has finished, enabling you to chain your scripts
- Managing central launches of all your scripts

However, difficulties remain in several areas:

- Managing scripts, evolutions, versioning, and so on
- Managing error checking

To address the general needs of running day-to-day routine procedures, most IT organizations incorporate batch schedulers. All teams (network, server, database, and such) plan their own operations to manage their services. These batch schedulers work well for all standard and routine procedures, but they lack the agility to evolve and dynamically answer to team or user requests. You would hardly align them in a cloud-oriented architecture!

Automation is the programmed execution of tasks within and between your applications. It involves processes such as BPA and ITPA.

## About Orchestration

Because IT teams are organized by their specialty and responsibilities, they are usually seen as silos, as Figure 1.2 illustrates.

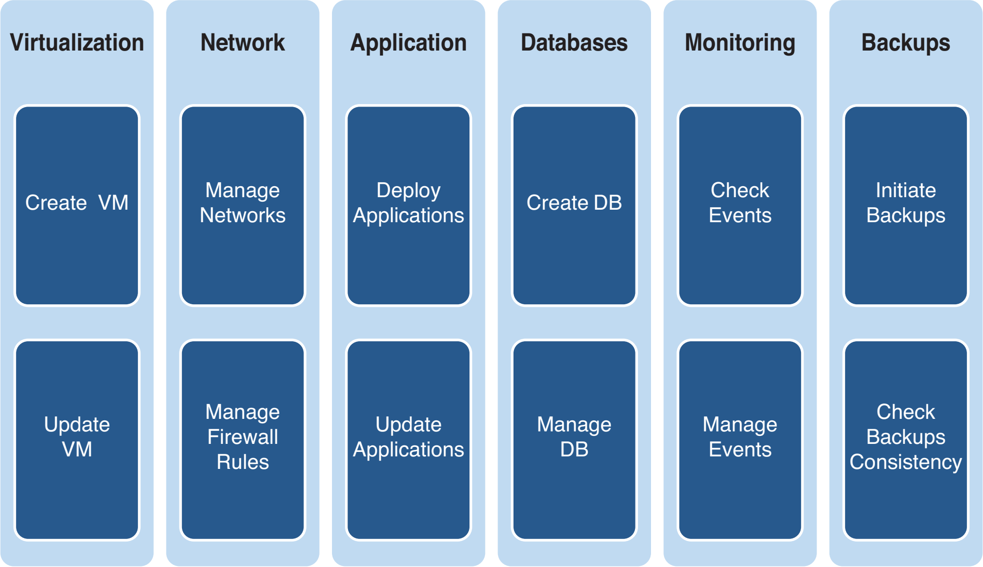


FIGURE 1.2 IT silos.

Here every team deals with its own business and has little communication with other teams. Processes are typically seen as inner-silo processes. This type of organization arises because every silo requires specific skills and specialists.

How would you deal with cross-silo processes in this organization? As a simple use case, consider the backup and reboot of an infrastructure dedicated to a specific web-oriented application. Here is the procedure:

- Stop monitoring.
- Create a temporary maintenance page for new users.

- Stop front-facing web servers and start backups.
- Stop application servers and start backups.
- Back up databases and then stop servers.
- Ensure that the backups occurred correctly, or raise an alarm.
- Restart the servers and applications in the right order (database, application, front ends).
- Check that each component is working properly.
- Restart monitoring.

Correctly completing this procedure requires dealing with many different teams and waiting for each team to develop its own process. The global procedure thus needs a considerable amount of time because of synchronization between teams—individuals from all teams must be available at the same time.

This is where orchestration helps, by organizing a full process across different silos and verifying that each step is complete before moving to the next. Orchestration is the external management of an application or applications. It can be thought of as the coordinator of a complex process, as Figure 1.3 shows. Orchestration ensures that all processes in different silos start at the correct time and behave as expected, and it conducts each process in the global process.

## Benefits of Orchestration

Using orchestration, you can take your organization to a new level with the capabilities to do the following:

- Conduct evolved cross-silo routine procedures
- Analyze and react to root causes of specific events across silos
- Create user- or business-oriented services based on sophisticated interaction between applications
- Save manpower in conducting procedures
- Shorten delays because all steps are synchronized

Orchestration enables you to address IT management through global and business-meaningful processes, regardless of what they are involved with technically.

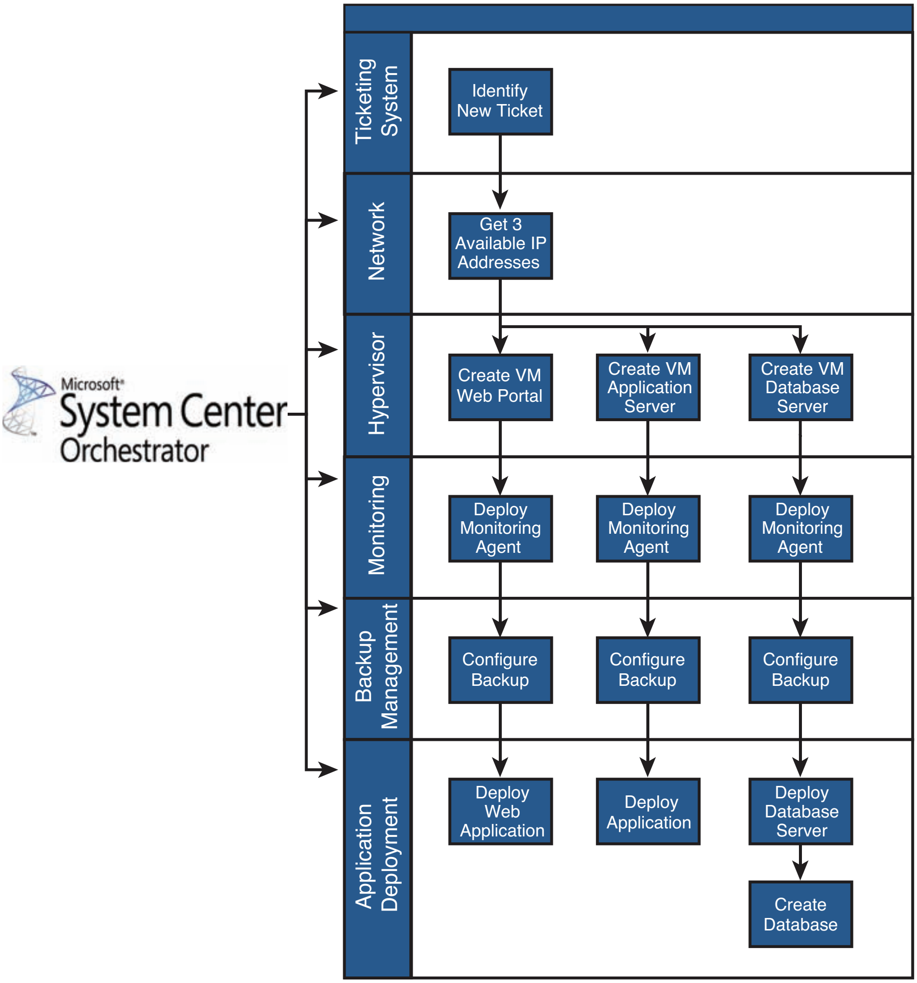


FIGURE 1.3 Orchestrating processes across silos.

What can you expect? The questions are always the same when you discover a new technology:

- What will it do for me?
- How easy is it to get access to this new technology?
- Is it worth it?

To help answer these questions, the next section discusses what you can expect from an orchestration project. Keep in mind that where your organization begins in its use of orchestration depends on a number of factors, including these:

- **Priorities:** These are your current issues and needs, along with your goals for the next several years.
- **Internal support skills:** Consider the state of your service or help desk.
- **ITIL implementation:** How far are you on the path to ITIL and how important is it to your organization? This includes policies for incident, change, and release management.

## How Orchestration Can Help

You can expect four types of benefits from an orchestration project:

- **Improved process quality and consistency:** You can consistently provide your users with the correct delivery process. Quality and consistency are often issues when delivering a procedure to an IT team. People tend to adapt the procedure to their individual needs, habits, or availability.

  When modeled in System Center Orchestrator, all your processes are handled in the same exact manner. This ensures consistency across all instances of a certain process.

- **Reduced routine workload and higher level of expertise for your team:** As you automate processes, you remove associated work from your team. This lets team members orchestrate their day-to-day routine tasks to focus on items of more value, such as event analysis, complex scenarios, new procedures, and runbook development.
- **Reduced delays of delivery to service requests:** Delays can be reduced in two areas:
  - **Capability to quickly consider a user request:** After an IT service is orchestrated, it can start processing user requests immediately and without waiting for someone to be available to address them. This means that the service can start as soon as a request is identified and mandatory resources are available.
  - **Synchronicity of all actions across silos:** In a classical human delivery model, delays arise between the time one team finishes its part of the job and the time the next team in the next silo begins its part.

  With tasks orchestrated, actions in all silos start synchronously. When the first series of tasks completes, the next set initiates. This ensures that the entire scenario is completed with the shortest possible delays, and without waiting between steps.

- **Capability to move to high request volumes and cloud compliancy:** When moving to automated service delivery, especially for cloud services, user requests increase dramatically. You cannot expect to manage these manually using a traditional approach. With orchestration, requests are addressed as soon as they occur or as soon as Information Systems can address them.

  At some point, you might face 10,000 user requests, yet your Information Systems group can handle only 100 concurrently. With orchestrated processes, you can narrow the number of requests treated in parallel to the maximum your infrastructure can handle.

### What Not to Expect with Orchestration

As discussed in the previous section, you can expect much from orchestration in all IT operations and cloud services. However, do not expect that starting an orchestration project or using System Center 2012 Orchestrator will deliver instant nirvana:

- Orchestrator does not define your processes. This is the first challenge you face after you decide to orchestrate your processes: Running Orchestrator does not instantly give you orchestrated processes.

  Processes, day-to-day operations, and IT services are your responsibility. System Center Orchestrator does not know anything about these areas and does not provide a methodology to organize them. It merely gives you the opportunity to automate and orchestrate those processes.

  This means that an orchestration project and a good implementation require you to understand your processes, define synchronization between tasks/steps, and check that these behave as expected.

- Orchestrator does not natively orchestrate everything. After seeing the integrated features, which are easy to use through a simple graphical interface, people might expect that they will find Orchestrator activities for everything. That is not the case.

  Activities in Orchestrator are interfaces to different systems, such as supervisory tools, configuration management, virtualization, databases, and enterprise resource planning (ERP) software. Although the System Center Orchestrator community is growing and is developing many integration packs (IPs) to extend Orchestrator functionality, third-party software organizations such as Hewlett-Packard and BMC Software must develop Orchestrator interfaces to their software.

  Connections exist to work with Microsoft management systems, such as all System Center 2012 components. Still needed are new and up-to-date interfaces to integrate with other recognized and widely implemented solutions such as HP OVO, BMC Remedy, and Nagios.

  Meanwhile, web services, PowerShell, and the command line are there to help you integrate your existing environment. You might even consider developing your own integration packs incorporating those commands.

## Positioning of Orchestrator in System Center 2012

System Center, a brand name for Microsoft's management platform, represents a means to integrate system management tools and technologies to help you with systems operations, troubleshooting, and planning.

The System Center 2012 product includes Operations Manager, Configuration Manager, Virtual Machine Manager, Service Manager, Data Protection Manager, Orchestrator, Endpoint Protection, App Controller, and Advisor. System Center Advisor, now offered at no additional cost to users of Microsoft server products, offers configuration-monitoring cloud services for Microsoft SQL Server, SharePoint, Exchange, Lync Server, and Windows

Server deployments; expect the list of monitored products to continue to grow. Microsoft's System Center 2012 cloud and data center solutions provide a common management toolset for your private and public cloud applications and services to help you deliver IT as a service to your business.

System Center builds on Microsoft's Dynamic Systems Initiative (DSI), which is designed to deliver simplicity, automation, and flexibility in the data center across the IT environment. Microsoft System Center components share the following DSI-based characteristics:

- Ease of use and deployment
- Based on industry and customer knowledge
- Scalability (both up to the largest enterprises and down to the smallest organizations)

Figure 1.4 illustrates the relationship between the System Center 2012 components and MOF.

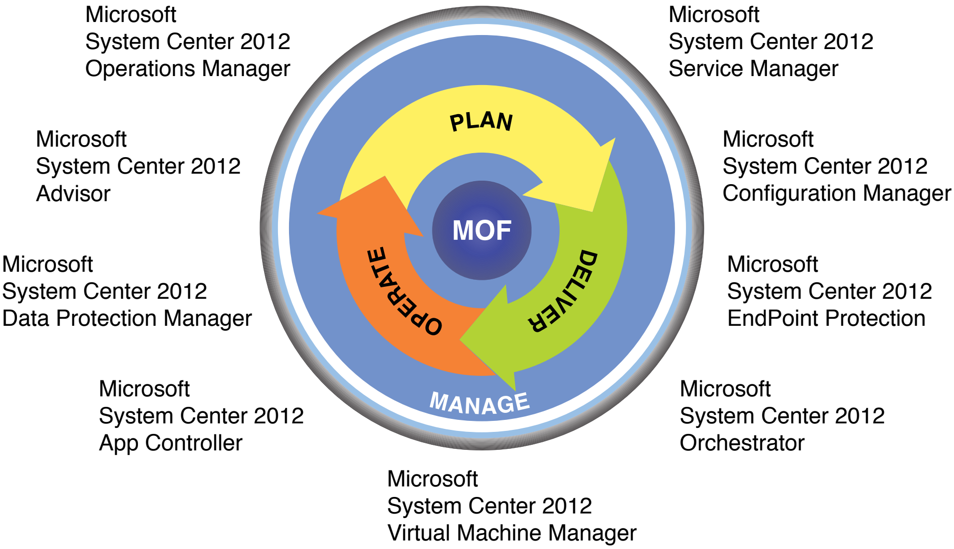


FIGURE 1.4 MOF with System Center components.

Orchestrator, the runbook automation component and the "glue" of System Center 2012, plays a central role in System Center, as Figure 1.5 shows. The right side of the figure shows the IT team, with the well-known System Center components: Virtual Machine Manager, Operations Manager, Configuration Manager, and Data Protection Manager. These are the typical tools you use to manage your infrastructure, whether it is physical, virtual, or already in a private or public cloud. Orchestrator is a powerful support tool for these IT teams because it helps in deploying and configuring, or supervising and administrating, the infrastructure.

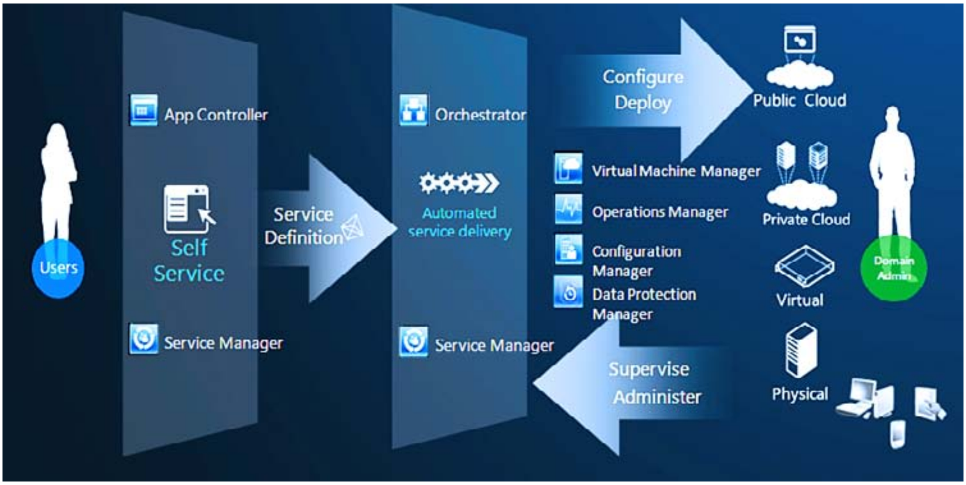


FIGURE 1.5 Orchestrator's role in System Center.

Orchestrator can be a great tool to automate all your IT configurations and deployments, such as when performing virtual machine (VM) creation and network integration; creating user accounts in Active Directory, Exchange, and third-party applications; or, depending on your infrastructure usage, moving a VM to a private cloud.

For operations, Orchestrator helps you supervise and administer your infrastructure. As an example, if all nodes from a SharePoint cluster are at about 80% utilization, you can use Orchestrator to create and integrate a new VM in the cluster. Orchestrator can also help you perform a server reboot by moving the target server to Operations Manager maintenance mode, protecting your SLA obligations before disconnecting users, starting a backup, and then restarting the server.

Integration with Service Manager becomes effortless as well. In fact, after your IT deployment and configuration processes are orchestrated, you can expose them to your entire IT team through a service catalog. You can easily set up this service catalog in Service Manager, which uses the System Center Service Manager IP to initiate the correct processes within Orchestrator. You can then take advantage of the entire service request lifecycle management within Service Manager. Orchestrator can update a request as closed if delivery or configuration was successful or, in case of errors, escalate the request to your team.

After your standard delivery and configuration processes are orchestrated and exposed, the next logical step is to expose those services to your users. Service Manager hides the technical complexity for users; you need to follow up on requests only when errors occur. This lets you offer services to your users at any time, without having to hire new people and manage them across time zones.

This architecture enables you to deliver those services to your users without day-to-day incremental work for your team, providing a high level of quality with very short delays.

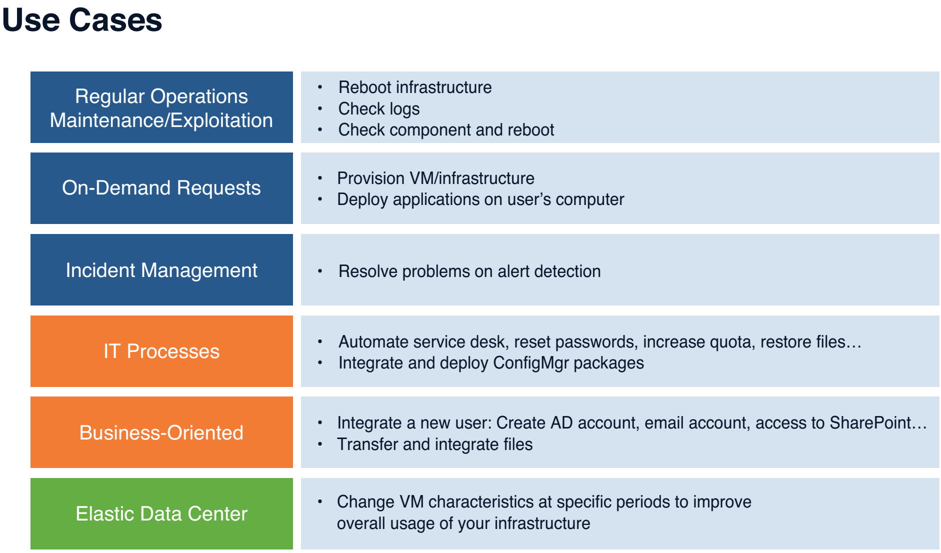
You can even get statistics on the number of requests from your users, average delivery delays, and so on. This is a very modern way to manage IT services and align them to business needs.

Finally, App Controller enables you to publish an application in a cloud environment, particularly Windows Azure, and obtain a global view about all related resources, wherever they are in the cloud or on your own IT environment.

Orchestrator plays a central role in System Center: It organizes and synchronizes all operations based on user requests, as well as events in your IT system. System Center Orchestrator helps you deliver the right service, with the shortest delay to your users, while also dramatically reducing the workload for your team.

# Typical Use Cases

Orchestration can accomplish a lot, but as with anything new in your IT toolbox, you must determine where to start. Figure 1.6 shows typical use cases you can implement in your organization to begin orchestration. The next sections discuss them in more detail.



Use Cases

| | |
|---|---|
| Regular Operations Maintenance/Exploitation | • Reboot infrastructure<br>• Check logs<br>• Check component and reboot |
| On-Demand Requests | • Provision VM/infrastructure<br>• Deploy applications on user's computer |
| Incident Management | • Resolve problems on alert detection |
| IT Processes | • Automate service desk, reset passwords, increase quota, restore files…<br>• Integrate and deploy ConfigMgr packages |
| Business-Oriented | • Integrate a new user: Create AD account, email account, access to SharePoint…<br>• Transfer and integrate files |
| Elastic Data Center | • Change VM characteristics at specific periods to improve overall usage of your infrastructure |

FIGURE 1.6 Typical use cases for System Center Orchestrator.

## Regular Maintenance and Daily Operations

IT requires regular performance of many operations to maintain healthy and efficient IT systems. These operations include deleting old temporary files, reindexing databases, and so on. To save time, you can easily create runbooks that will perform those operations for you at the proper time.

A typical example is an application infrastructure reboot that does the following:

- Waits for a maintenance period
- Sets servers in maintenance mode
- Disables new user connections
- Disconnects all connected users
- Shuts down front-end servers
- Starts application server backups and ensures backup integrity
- Shuts down application servers
- Starts database server backups and ensures backup integrity
- Reboots database servers
- Starts application and then front-end servers
- Checks that the application is running correctly
- Re-enables user connections
- Takes servers out of maintenance mode

Because those processes deal with only IT management, you can define them easily with your team. The production implementation is free of risk because processes are triggered and reported to your team only. These types of projects are easy to accomplish with System Center Orchestrator and are valuable for your team.

### On-Demand Requests

You might have operations to perform on request that are not as regular as your daily operations. In this case, you wait for an external signal to trigger a complex process. You typically need to respond to the request as quickly as possible.

This could be delivery of IT resources such as VMs, shares, restore operations, application deployment, a new server in a Configuration Manager collection, and so on. In these cases, you can wait for events to appear and then trigger runbooks yourself, or you can expose them as a service catalog in Service Manager.

### Incident Management

Incidents differ from problems from an ITIL viewpoint because you know how to recognize them and then respond to them. Even better, you can easily create runbooks that check your systems for events related to an incident. This is accomplished using logs or Windows event log checking, events in Operations Manager, and so on. You can even correlate events from different sources to identify an incident. When an incident is identified, you can automate corrective actions such as restarting an application or a service, deleting temporary files, and such.

You can extend this principle of analysis and corrective action to any type of event and corrective action, particularly if you need to interact with different systems.

## IT Process Automation

IT process automation typically consists of requests from users to IT teams. You can identify them easily with a service desk team through an analysis of user requests. Here are some examples:

- Unlock a user account
- Reset a password
- Increase disk quota
- Restore a file from a backup
- Deploy an application on a user computer

Here automation can save your team a considerable amount of time and help deliver user services more efficiently.

## Business-Oriented Processes Automation

Those processes are extensions of IT processes and focus on the organization's activity. Here you are looking for increases in company productivity rather than IT increases in efficiency.

This could be all the tasks necessary to set up a new user: creating an AD account, creating an Exchange account, setting up a storage resource in the data center nearest where the user will be located, providing access to SharePoint folders based on the user's job, and so on. In addition, you must delete all the user's accesses when he or she leaves the organization.

Your company could have orders coming from different stores worldwide. Order integrity can be checked using a specific tool. After validating, you can pass the order to another application that integrates all orders in your production system. These types of applications can be quite useful if your company has many offices worldwide. You simply ensure that all orders or information are treated exactly the same, regardless of where they come from. Alternatively, you can imagine delivering information or triggering operations to many remote locations.

## Elastic Data Center

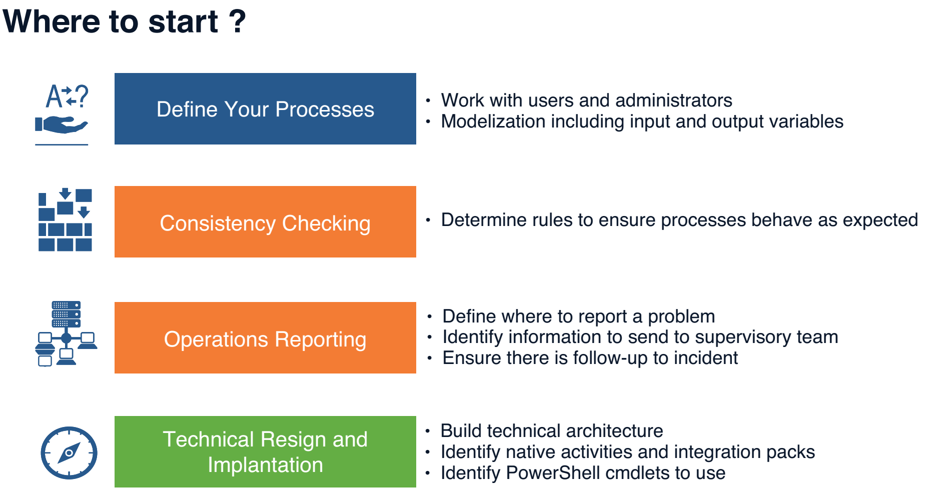
Increasing the size of your data center is generally easy, from a technical point of view (although not necessarily in terms of budget). However, you might be able to better utilize resources by provisioning them when needed and releasing them when they are no longer required.

For example, in a data center, some servers are frequently used during peak times and other servers are used only in nonpeak periods. Using Orchestrator, you can reduce virtual

CPUs and memory during nonpeak times and increase those resources during peak times. This lets you adjust your data center usage more efficiently, based on resources needed at certain times.

# Managing a Project

Beyond any typical use cases, it is important for you is to be able to identify projects within your organization and orchestrate them. Four key steps can make your orchestration process a success. Figure 1.7 displays these steps, and the following sections cover them in more detail.



Where to start ?

Define Your Processes
- Work with users and administrators
- Modelization including input and output variables

Consistency Checking
- Determine rules to ensure processes behave as expected

Operations Reporting
- Define where to report a problem
- Identify information to send to supervisory team
- Ensure there is follow-up to incident

Technical Resign and Implantation
- Build technical architecture
- Identify native activities and integration packs
- Identify PowerShell cmdlets to use

FIGURE 1.7 Steps in an Orchestrator project.

## Define Your Processes

You first must reach an agreement with your users on the service you will deliver. What is the purpose of your process? Will it improve quality of your delivery, reduce delays for users, and/or save you time? Remember that you are targeting users' processes, meaning that you must deliver the service they expect; otherwise, your orchestration will be useless.

After identifying your process, describe it. Include all the steps and interactions with infrastructure, applications, and so on. Then define inputs needed from users and results, as well as feedback to send them. Microsoft Visio can be a very useful tool for this task.

Even if you are comfortable with what you want to orchestrate, the authors encourage you to describe the process to identify interactions with users as well as your team. For example, VM generation is a simple process, but integrating the VM with the right task sequence and according to business usage is more complex.

### Consistency Checking

For every process, determine how to ensure that it is executed correctly. In this step, you reproduce what people working in teams are doing: One person executes an operation, and another team member checks the operation's results.

For instance, you can check execution details of your process, such as whether the VM you created has the expected number of virtual processors, memory, and hard drives. Another example is checking the number of files a program generates. When sharing files between systems, check the number of files to transfer from one location to another and the associated amount of data.

### Report on Operations

Now that you have identified rules to ensure that your process has executed, consider whether the results are as expected. You must report this fact to the appropriate person or team. A series of simple questions can help determine this:

- What should be reported? All execution details? Only warnings and errors?
- Who should be warned? User, supervision team, application team, someone else?
- What contextual information must be sent to understand what is occurring?
- How do you ensure follow-up for errors?

### Technical Implementation

At this point, your process is well defined, you are able to describe the rules to identify a correct behavior or an unexpected one, and you know how to send warnings/alarms if problems arise. You can now implement your process within Orchestrator. This book helps you do so in the right manner with, for example, Chapter 9, “Standard Activities,” and Chapter 13, “Integration with System Center Operations Manager” through Chapter 18, “Integration with Windows Azure,” is your reference to interact with all System Center components and Windows Azure. Chapter 12, “Orchestrator Integration Packs,” introduces you to other activities available through the other available IPs, extending the standard activities available in System Center Orchestrator.

## Example: Orchestrating a VM Deployment

The example in this section demonstrates the different steps in orchestrating a process. In this example, a virtual machine is created and configured upon user request.

Figure 1.8 describes the process details of delivering this VM with roles defined in the user request and ensuring that all elements are up-to-date and configured as expected. Here's how it works:

- A user request in Service Manager triggers the process. After the request is validated, the information is stored and Orchestrator can identify this request later (such as at night, when less activity occurs).

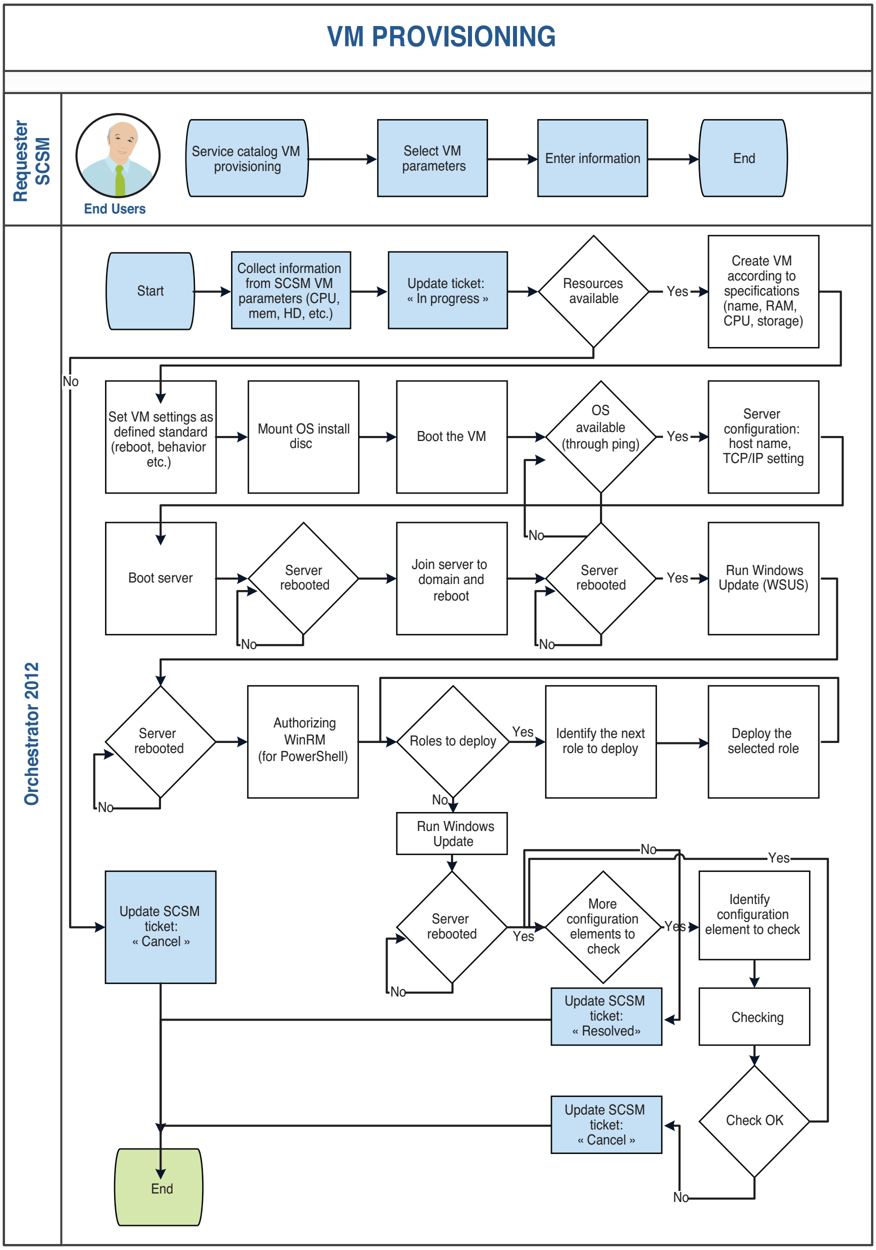

VM PROVISIONING
Requester SCSM
End Users
Service catalog VM provisioning
Select VM parameters
Enter information
End
Orchestrator 2012
Start
Collect information from SCSM VM parameters (CPU, mem, HD, etc.)
Update ticket: « In progress »
Resources available
Yes
Create VM according to specifications (name, RAM, CPU, storage)
No
Set VM settings as defined standard (reboot, behavior etc.)
Mount OS install disc
Boot the VM
OS available (through ping)
Yes
Server configuration: host name, TCP/IP setting
No
Boot server
Server rebooted
No
Join server to domain and reboot
Server rebooted
No
Yes
Run Windows Update (WSUS)
Server rebooted
No
Authorizing WinRM (for PowerShell)
Roles to deploy
Yes
Identify the next role to deploy
Deploy the selected role
No
Run Windows Update
No
Yes
Update SCSM ticket: « Cancel »
Server rebooted
No
Yes
More configuration elements to check
Yes
Identify configuration element to check
Update SCSM ticket: « Resolved»
Checking
Update SCSM ticket: « Cancel »
Check OK
No
End

FIGURE 1.8 VM delivery process.

- Before creating anything, the Orchestrator process checks whether resources are available to create the machine, particularly whether enough storage is available.
  - If resources are available, the VM is created within Virtual Machine Manager (or any virtualization software) with the requested parameters (CPU, memory,

disk size, and other specifications). The OS install disk is mounted, and the VM is booted.

- If resources are insufficient, the Service Manager ticket is updated to Cancel status.

- When the server answers to a ping request, it is considered available and its server configuration can be updated. After a new reboot, the server can join a target production domain. After the domain join, the server is rebooted again and WinRM is authorized, enabling you to remotely run PowerShell on the VM.

  Roles can now be deployed and configured using PowerShell commands from System Center Orchestrator. After all roles are deployed, another Windows update is performed through Windows Software Update Services (WSUS).

- When all configuration actions are completed, the last part of the process is to confirm that everything is configured as expected. If this is the case, the ticket status is updated to Resolved; otherwise, it updates to Cancel.

This process shows that many operations surround the creation of a simple VM to manage the correct configuration of roles and integration in your network and domain. Based on your security rules, you might have even more steps.

Here's how this can be translated to Orchestrator runbooks:

As Figure 1.9 shows, the overall process is rather short and triggers different runbooks (Resources Available, Create VM-Server, Roles to Deploy, DNS Roles to Deploy, Hyper-V Roles to Deploy, AD Domain Server Roles, Configuration Check) to execute complex technical operations. Those runbooks can be considered as procedures.

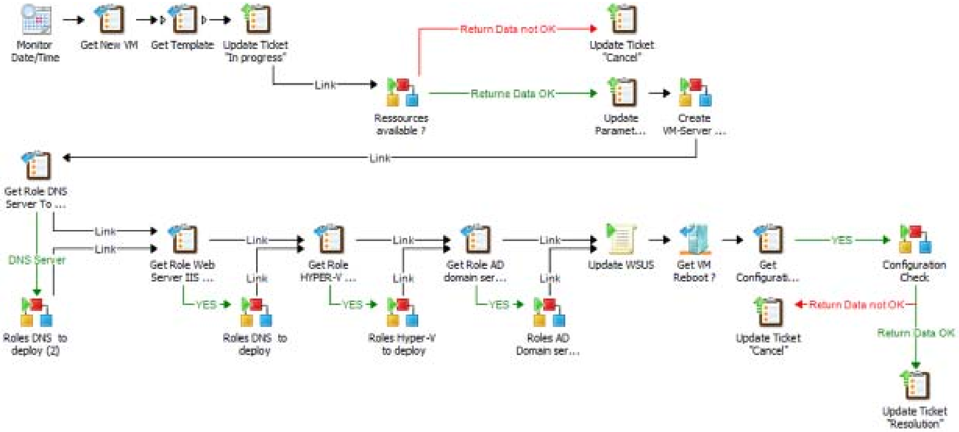


FIGURE 1.9 Runbook showing overall VM delivery process.

The process in Figure 1.10 checks whether enough space is available on any of the virtualization host data stores and returns the correct store to use. If space is insufficient, the return value is zero and the global VM Delivery process stops.

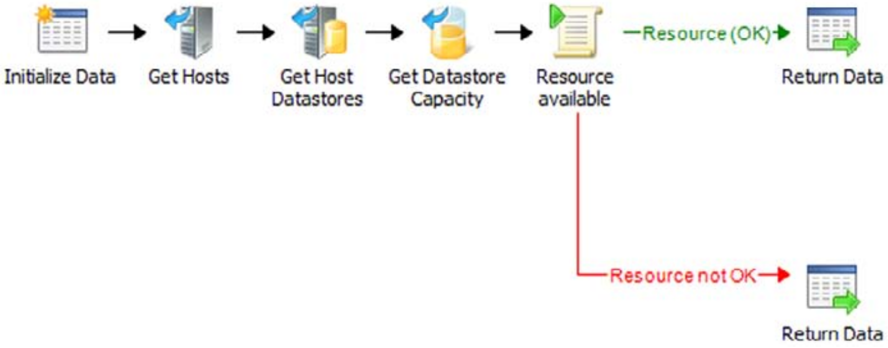


FIGURE 1.10 Check whether enough storage is available.

Figure 1.11 shows the runbook creating a new VM from a template. After this VM is created, it starts and the network configuration is updated. The next step is to join the server on the right domain, which is validated by a system reboot. The system is then updated through WSUS before WinRM is authorized (enabling PowerShell execution remotely).

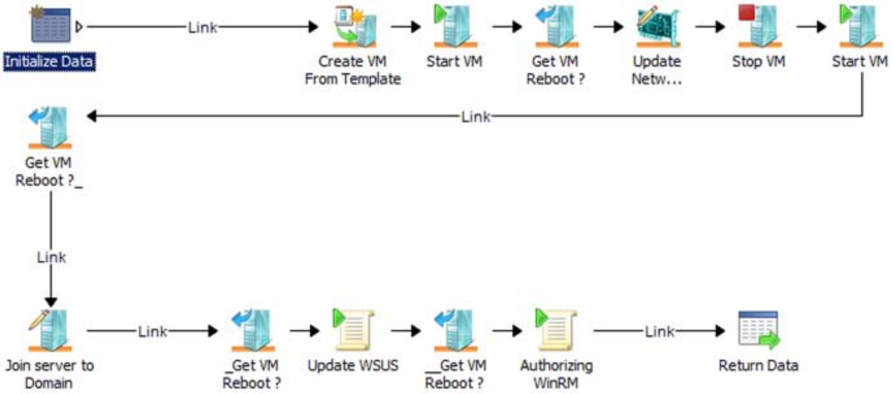


FIGURE 1.11 Create and update the virtual machine.

The VM is now up-to-date, in the correct domain and network, ready to accept PowerShell commands. Deployment of a new role on the server is straightforward, as Figure 1.12 describes. The role to deploy is passed as a parameter to the **Initialize Data** activity. This parameter is used in the PowerShell Deploy Select Role script that actually activates the role. The request ticket is then updated to reflect that the role has been created as requested on the server.

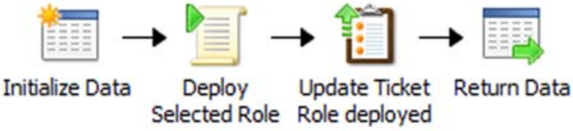


FIGURE 1.12 Roles deployment on the server.

A series of checks is performed on the VM to ensure that the configuration is as expected, as Figure 1.13 shows. This runbook then returns a status to the global process.

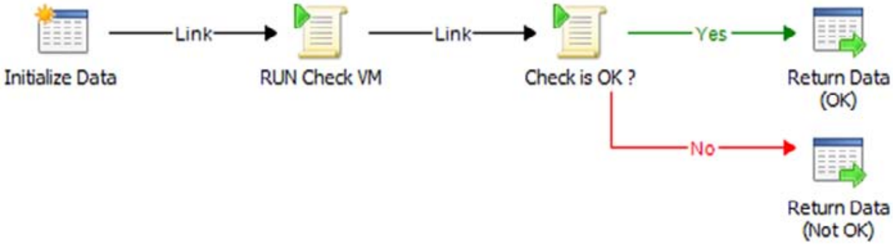


FIGURE 1.13 Check that VM configuration is as expected.

## Summary

A System Center Orchestrator project can present great opportunities to improve and stabilize the quality of service of your team and improve responsiveness to user requests.

Here's what you can expect by using Orchestrator:

- Increased stability in your operations, with reduced errors
- Improved productivity through delegated operations to System Center Orchestrator
- Shortened delays for delivery because process are executed without waiting between operations, as with a traditional approach and teams

From an organizational prospective, Orchestrator projects require that you have a global view of IT operations and how different IT teams execute these operations.

This chapter introduced System Center Orchestrator, the integration, automation, and orchestration component of System Center 2012. The next chapter discusses changes in this release.

*This page intentionally left blank*

CHAPTER 2

# What's New in System Center 2012 Orchestrator

IN THIS CHAPTER

- ▶ The History of Orchestrator
- ▶ OIS 6.3 Versus Orchestrator 2012

In its second major release since its acquisition by Microsoft, Orchestrator (previously known as Opalis Integration Server, or OIS) has completed its assimilation into System Center. Chapter 1, "Orchestration, Integration, and Automation," introduced the concepts behind run book automation (RBA), business process automation (BPA), IT process automation (ITPA), and Orchestrator. This chapter focuses on changes to Orchestrator in System Center 2012. If you have an OIS 6.3 background, reading this chapter can provide a smooth transition to understanding this System Center component. The chapter covers technology changes and discusses how Microsoft's rebranding affects Orchestrator's position in System Center. This chapter also provides a brief overview of the history of Orchestrator.

As the first version developed entirely by Microsoft, System Center 2012 Orchestrator has the benefit of the rigorous testing and code standards placed on all Microsoft products. In addition, it has the benefit of several years of experience with customers implementing OIS into their data centers; Microsoft has taken that feedback and fed it into product development. Although the user interfaces for Orchestrator are similar to the previous version, they have received a facelift along the lines of the rest of the System Center components, providing a consistent look and feel across the product.

The underlying theme is that even though Orchestrator appears different and has a new name, the technologies, concepts, and processes underneath essentially remain the same. In fact, this version further emphasizes the features and benefits of OIS 6.3. Integration is still what Orchestrator is about, and it continues to offer the same robust workflow engine. If you used the last release of OIS, System Center 2012 Orchestrator will be a familiar experience. With that said, you will encounter some key terminology changes, new software and hardware prerequisites, several dropped features, and a brand-new Orchestration console.

## The History of Orchestrator

Orchestrator has had a relatively short life in the hands of Microsoft, but its predecessors by Opalis Software, Inc., hit the shelves more than a decade ago. Opalis Software enjoyed a successful run, and its history includes a number of milestone developments that helped shape what Orchestrator is today. Even in the first release of the OpalisRobot product, the company approached automation differently from the rest of the world. Simply scheduling jobs was not enough; the real value was in being able to monitor for certain events and use those to trigger an action. By combining low-level task automation with the capability to integrate heterogeneous tools, people, and processes, Opalis enabled much more consistent and reliable automation. This concept came to be known more formally as IT process automation. The following sections look at how Orchestrator came to be and examine the advancements Microsoft has made since the 2009 acquisition.

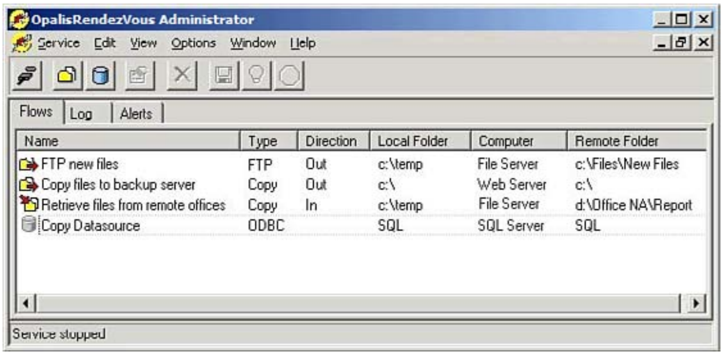
### The Beginnings of Orchestrator: OpalisRobot

Orchestrator started life in 1995 as a program called OpalisRobot; Figure 2.1 shows the Opalis logo. As OpalisRobot evolved over the next decade, it became clear it had an important differentiating feature over its competitors: Whereas other products were essentially task schedulers, OpalisRobot incorporated monitors and triggers. The idea was not only to schedule automated tasks, but also to dynamically identify and respond to specific events in your environment. This enabled administrators to build truly self-healing systems and applications. This concept was a precursor to runbook automation, and it is still very much at the core of Orchestrator today.

FIGURE 2.1 Opalis logo.

### OpalisRendezVous

OpalisRobot was not the only product Opalis Software developed and produced. The company also sold OpalisRendezVous, which provided a graphical user interface (GUI) for transferring files over FTP, file shares, and databases. This product offered a unique "when, what, where" configuration that enabled administrators to control the flow of file distribution, ultimately allowing a company to move quickly from a manual to an automated process. Again, simplicity of use was an underlying principle that made OpalisRendezVous such a useful and popular product. Figure 2.2 shows the OpalisRendezVous interface.



| Name | Type | Direction | Local Folder | Computer | Remote Folder |
|---|---|---|---|---|---|
| FTP new files | FTP | Out | c:\temp | File Server | c:\Files\New Files |
| Copy files to backup server | Copy | Out | c:\ | Web Server | c:\ |
| Retrieve files from remote offices | Copy | In | c:\temp | File Server | d:\Office NA\Report |
| Copy Datasource | ODBC |  | SQL | SQL Server | SQL |

Service stopped

FIGURE 2.2 OpalisRendezVous user interface.

## Opalis Innovates

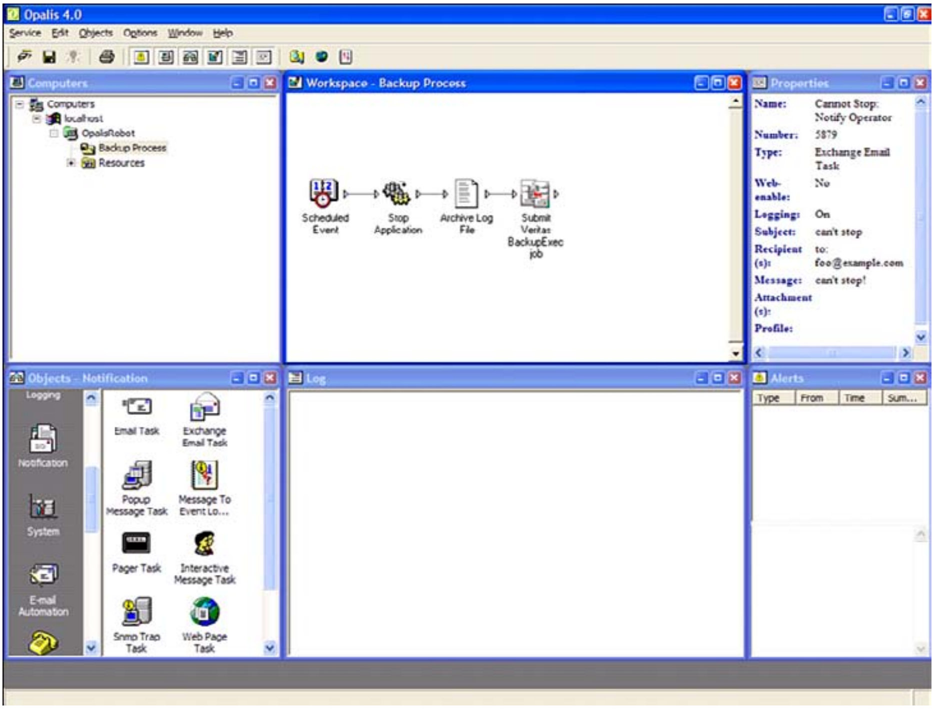
OpalisRobot 3.0 was released in 1997, bringing one of the most important innovations to the product line with the world's first drag-and-drop design interface for workflows. This was an important development because it marked a key concept that exists in current System Center products: simplicity. Ease of operation and administration has been an important theme throughout all System Center components.

A year later, Opalis released a set of add-ons for email and computer telephony integration. These add-ons, today called *integration packs* (IPs), facilitated the addition of activities to the set of out-of-the-box activities shipping with the product. Over the years, Opalis fostered a community of independent developers to create open source IPs that enable the product to automate tasks within many other systems. These IPs changed the perception of OIS from an ITPA tool separate from the rest of the data center to that of a platform resting beneath all the tools and processes in the data center.

This important distinction led to what is now known as the Orchestrator Integration Toolkit. It enables developers to integrate Orchestrator with virtually every other application, regardless of manufacturer, through those other applications' exposed integration surfaces, such as application programming interfaces (APIs), command-line interfaces (CLIs), and databases.

Microsoft currently offers more than a dozen supported IPs for both Microsoft and other vendor applications, such as VMware vSphere and HP Service Manager. Dozens more are available through open source community developers.

OpalisRobot 4.0, released in 2002, was the last release under the OpalisRobot brand. This final release brought a new user interface (see Figure 2.3), some bug fixes, and additional standard automation objects. This release was also the first with support on Linux and Solaris; however, support on non-Microsoft platforms ceased with 4.0 and did not carry forward to later versions of the product.



Name: Cannot Stop: Notify Operator
Number: 5879
Type: Exchange Email Task
Web-enable: No
Logging: On
Subject: can't stop
Recipient(s): to: foo@example.com
Message: can't stop!
Attachment(s):
Profile:

FIGURE 2.3 OpalisRobot 4.0 interface.

## Goodbye Robot, Hello OIS

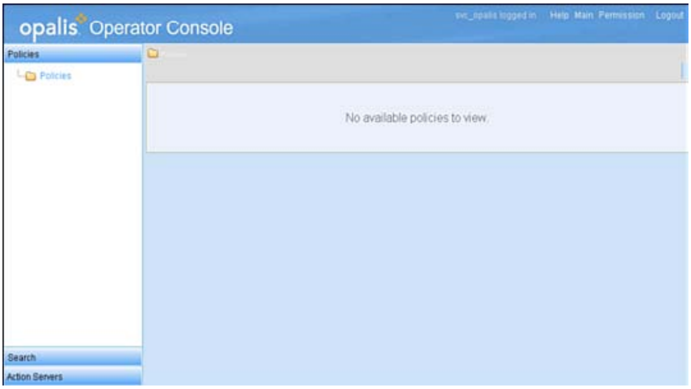
By the early 2000s, it became clear that although Opalis Software clearly understood where it needed to fit into runbook automation and ITPA, OpalisRobot had outgrown its architecture; it was time for a major rewrite of the underlying technology. Opalis retired its RendezVous and Robot product lines and planted its position firmly in the ITPA space. Fundamentally, this was a shift in focus, from developing better runbook activities to providing a better integration platform. New integration packs (then called connector access packs) were released to support this positioning, which included integration into Microsoft Operations Manager.

As part of this new positioning, Opalis rebranded its new automation software as Opalis Integration Server and released OIS 5.0 in 2005. OIS 5.0 brought a round of significant improvements, including the use of an industry-standard relational database management system on the back end, dashboards, improved scalability, and Active Directory integration. The marriage of the administrator-friendly interface, the IP approach, and the new architecture allowed OIS to take its seat as a true ITPA tool, allowing automation of activities to occur across systems and processes.

Issues with the redesigned architecture became evident over the following months, as often occurs with newly released software. Opalis made several incremental improvements to the 5.x release, and those ultimately led to the development of a new workflow engine, called *pipeline mode*. Pipeline mode changed how data was passed between objects, facilitating new capabilities such as embedded looping and the capability to flatten published data. The old workflow engine, referred to as *legacy mode*, remained available until the System Center 2012 Orchestrator release. A final round of minor changes brought about the last major release of OIS with version 6.0.

## Microsoft's Acquisition of Opalis Software

Microsoft, having identified a requirement to bolster its line of data center management tools with an ITPA solution, acquired Opalis Software in December 2009. The terms of the acquisition included a final release of OIS for Microsoft that removed any unacceptable features, such as the Java-based prerequisite of the OIS Operator Console displayed in Figure 2.4. For legal reasons, Microsoft would not distribute the open source software required for the Operator Console. However, the console itself was still available and supported until Orchestrator was released as part of System Center 2012 in April 2013.



No available policies to view.

FIGURE 2.4 OIS Operator console.

**NOTE: EXISTING CUSTOMER CONSIDERATIONS POST-ACQUISITION**

When Microsoft incorporated OIS into its existing System Center licensing, it offered a grant of Server Management Suite Datacenter (SMSD) licenses to existing customers to the monetary equivalent of their lifetime purchases with Opalis Software, as long as they purchased a two-year Software Assurance contract. Opalis Integration Server was the only product Opalis Software offered at the time of the acquisition, so Microsoft continued development of all Opalis software products. The Opalis Dashboard, sold by Opalis Software but developed by Altosoft, was available directly from Altosoft for a period of time, but it has since been discontinued.

Microsoft positioned the Opalis software under System Center. Version 6.3, which was the final update to OIS, included support for OIS on Windows Server 2008 and the OIS Client on Windows 7, and a set of IPs for System Center. Figure 2.5 shows the OIS 6.3 Client.

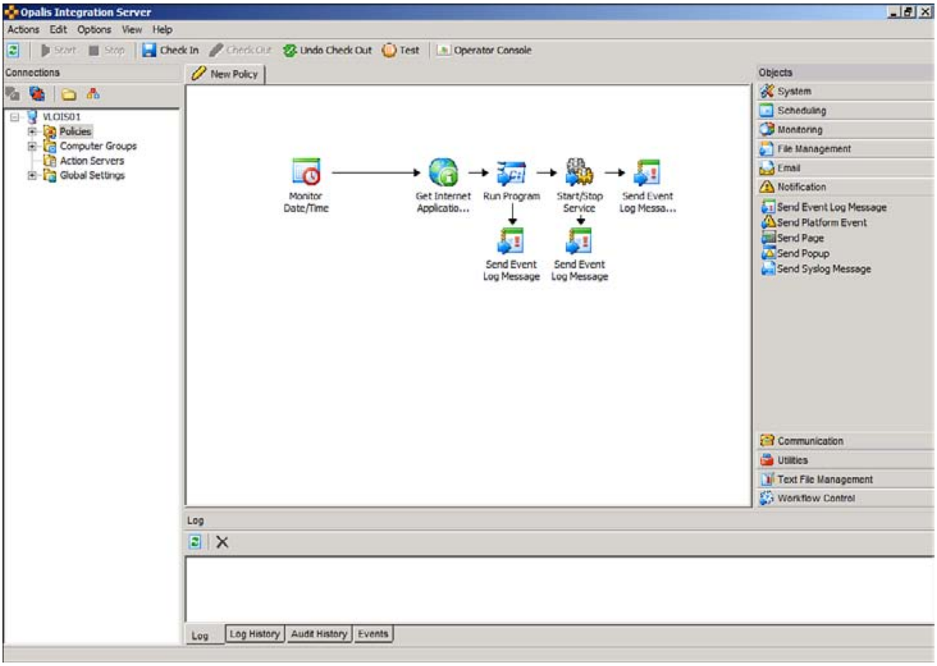


FIGURE 2.5 OIS 6.3 Client.

## OIS to Orchestrator

Microsoft announced the rebranding of Orchestrator in March 2011 at the Microsoft Management Summit in Las Vegas. Officially called System Center 2012 Orchestrator, this is the first major release developed wholly by Microsoft. As such, the functionality is migrated into a Microsoft codebase. This means that Orchestrator is now subject to the same rigorous design and testing cycles as the rest of the Microsoft products.

With the System Center 2012 release, OIS 6.3 was no longer available as a standalone download, but Microsoft provided support of the product for an additional 12 months. The company also honored existing support agreements with customers.

Orchestrator brings a series of improvements, including these:

- Bug fixes
- Terminology changes
- A new Orchestration console
- Updated integration packs
- A new installer

**NOTE: NEW WITH ORCHESTRATOR 2012 SERVICE PACK 1 AND R2**

System Center 2012 Service Pack 1 changes to Orchestrator include:

- New integration packs (Exchange Administrator, Exchange Users, FTP, and Representational State Transfer, or REST)
- Updates to the Active Directory, HP Service Manager, VMware vSphere, System Center 2012 Operations Manager, and System Center Virtual Machine Manager 2012 integration packs
- Support for the Windows Server 2012 and SQL Server 2012 platforms

See http://technet.microsoft.com/en-us/library/jj614522.aspx for information.

Changes to Orchestrator in System Center 2012 R2, in pre-release when this book was printed and documented at http://technet.microsoft.com/en-us/library/dn251064.aspx, include:

- Support for Windows Server 2012 R2
- Changes to the installation program to install the Service Management Automation web service and up to three runbook workers
- A SharePoint integration pack
- Updates to the Windows Azure and Virtual Machine Manager 2012 IPs

## OIS Migration to Orchestrator

You cannot upgrade OIS to Orchestrator, but you can migrate existing OIS 6.3 policies to Orchestrator 2012. Some of the standard activities have changed, so you might need to adjust your runbooks after migrating them from OIS 6.3. Chapter 5, “Installing System Center 2012 Orchestrator,” covers Opalis migration in detail.

## Where Orchestrator Fits into System Center

Microsoft has positioned System Center 2012 as a single product with multiple components rather than individual applications, which is representative of the way the tools interact with each other. The components have a high level of integration, and Orchestrator is key to that integration. This integration also reflects the license options: System Center 2012 has a single SKU with an option to purchase either licenses per virtual machine (VM) or an unlimited VM enterprise license. Figure 2.6 illustrates the relationships among the different System Center components.

Microsoft built System Center 2012 to manage on-premise, private cloud, and public cloud data centers. Each component provides a platform; on top is a set of solutions that fulfill those management needs. Here is a description of each component—see http://technet.microsoft.com/en-us/library/hh546785.aspx for additional information:

- **App Controller:** Enables template-based deployment of services and virtual machines to private clouds via Virtual Machine Manager and public clouds using Windows Azure.

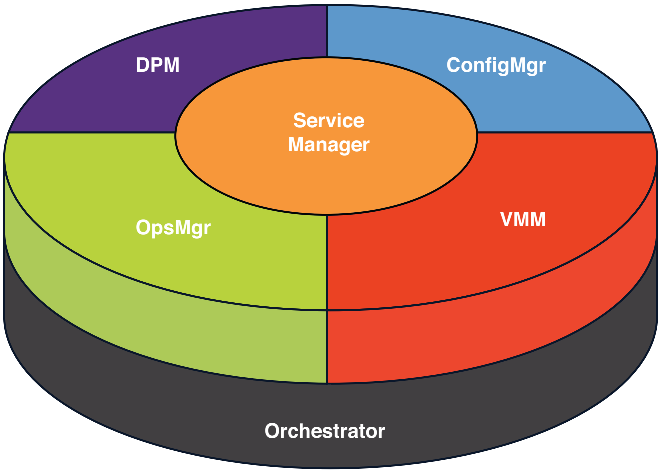


FIGURE 2.6 IT management as a platform.

- **Configuration Manager:** Provides a comprehensive configuration management solution for the Microsoft platform. This component features application delivery, operating system deployment, desktop virtualization, device management, compliance monitoring and remediation, hardware monitoring, and software inventory capability.
- **Endpoint Protection:** Endpoint Protection is built on the Configuration Manager platform and provides antimalware and security solutions. Because it shares its infrastructure with Configuration Manager, you can consolidate endpoint protection and management.
- **Data Protection Manager (DPM):** DPM is a centralized backup solution that features near-continuous backup. It enables rapid and reliable recovery of a Windows environment, including Windows servers and desktops, SQL Server, Exchange Server, and SharePoint.
- **Operations Manager:** Provides an infrastructure management solution that delivers comprehensive health and performance monitoring and alerting to drive performance and availability for data center and cloud-based applications.
- **Orchestrator:** Enables the automated delivery of IT services through a simple user interface that is built for information technology (IT) administrators. Orchestrator enables automation across a heterogeneous datacenter.
- **Service Manager:** Provides a platform for managing Microsoft Operations Framework (MOF) and IT Information Library (ITIL)–based service management processes. These include incident management, request fulfillment, problem management, change management, and release management. Those processes are automated through integration with companion System Center 2012 components.

- **Virtual Machine Manager (VMM):** VMM is a virtual infrastructure management solution for provisioning and centrally managing host, network, and storage resources that support datacenter, private, and public cloud environments.

Orchestrator is unique, in that it does not provide a solution to any problem; it provides a platform and set of activities to enable administrators to generate their own solutions to unlimited problems. Often the question with Orchestrator is not whether you *can* automate something, but whether you *should* automate it. Automation clearly has many benefits, but a certain level of planning must go into the design and creation of runbooks. The good news is that Orchestrator simplifies this process with its user-friendly Runbook Designer.

Orchestrator shines particularly well in the following areas:

- Automation in the data center
- Service delivery and automation
- Creation of self-healing systems

The best way to think of Orchestrator is not as an additional component hanging off the end of the rest of System Center, but one sitting beneath the rest of the components that can read, interact with, and pass data among the various APIs to act as a point of integration. In this way, Orchestrator doesn't necessarily need to action all the automation, but it can act as a puppet master that enables other applications to execute the automation.

# OIS 6.3 Versus Orchestrator 2012

On the surface, certain areas of Orchestrator 2012 appear to differ greatly from the OIS 6.3 release, but the underlying concepts and processes remain relatively the same. All user interfaces have had facelifts, and the OIS Operator Console has been completely rebuilt from scratch.

The next sections discuss these changes and include a brief overview of the features that were improved or rebuilt. Additional detail about each of these features and their uses and configuration options is available in Chapter 3, "Looking Inside System Center 2012 Orchestrator," and Chapter 4, "Architectural Design."

## Terminology Changes

Thanks to rebranding and the Microsoft acquisition, several terms have changed between OIS 6.3 and Orchestrator, but much parity exists between the legacy and the new Orchestrator features. Some pieces, such as the License Manager, were removed altogether; others, such as the Orchestration console, were rebuilt from the ground up. In general, however, the interfaces and features in Orchestrator should be familiar if you have used OIS 6.3. Table 2.1 lists the terminology changes within the architecture features.

TABLE 2.1 Feature Terminology Changes

| OIS 6.3 | Orchestrator 2012 |
|---|---|
| SQL Data Store | Orchestrator Database |
| Opalis Management Server | Orchestrator Management Server |
| Opalis Action Server | Orchestrator Runbook Server |
| OIS Client (Authoring Console) | Runbook Designer |
| Policy Testing Console | Runbook Tester |
| OIS Operator Console | Orchestration Console |
| Deployment Manager | Deployment Manager |
| OIS Web Service (WSDL) | Orchestrator Web Service |
| Database Configuration Utility | Data Store Configuration |
| License Manager | — |

### Orchestrator Database

A Microsoft SQL Server database stores all data and configurations. This database is a critical feature and should be configured for high availability. If the SQL Server goes down, runbook servers cannot execute any runbooks. Orchestrator uses one database with a default name of Orchestrator and a correlation of SQL_Latin1_General_CP1_CI_AS.

**NOTE: ORACLE DATABASE SUPPORT**

Support for Oracle as the relational database management system (RDBMS) is not included in Orchestrator, as it was in Opalis Integration Server.

### Orchestrator Management Server

The management server exists primarily to establish communication between the design features and the SQL database. It is not a critical runtime feature and does not necessarily need to be highly available. This feature fills the same role as the OIS management server in the previous release.

### Orchestrator Runbook Server

The Orchestrator runbook server is the feature that actually executes runbooks. You can deploy multiple runbook servers to allow for load balancing. This feature handles the same responsibilities as the action server in the previous release.

### Runbook Designer

The Runbook Designer console is used to design, test, and implement all runbooks. This feature is not critical to the operation of existing runbooks and, therefore, does not necessarily need to be highly available. This feature is essentially the same as the OIS 6.3 Client.

### Runbook Tester

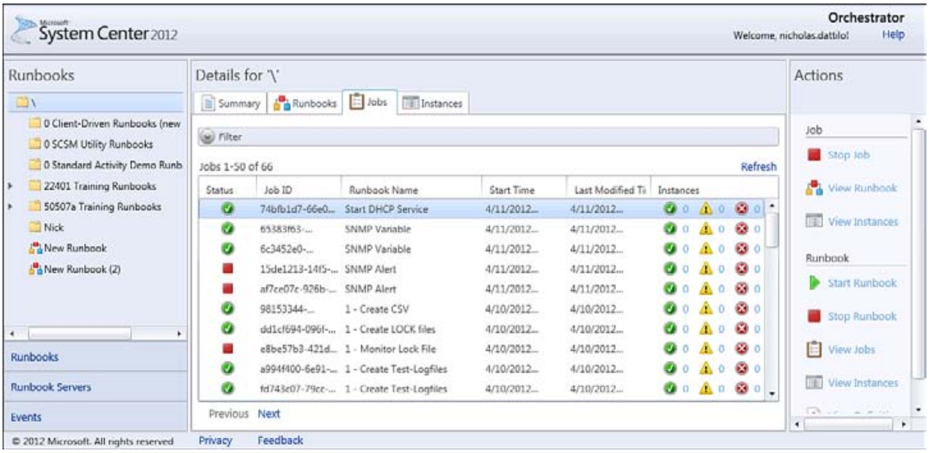
The Runbook Tester, which is launched within the Runbook Designer, has a similar function and layout to the OIS 6.3 Policy Testing console. This tool is used to test runbooks before deployment and publishes runtime data about each activity as the runbook steps through from beginning to end.

**CAUTION: RUNBOOK TESTER COMMITS CHANGES**

Several times throughout this book, the authors state that the Runbook Tester actually executes and commits changes when testing a runbook. It does not display "what if" data or scenarios. Keep this in mind, and use a development environment whenever a runbook might affect existing IT services.

### Orchestration Console

This console, displayed in Figure 2.7, provides IT operators with a thin-client interface into Orchestrator. The Orchestration console is not critical to the runtime of runbooks, but it enables users to view the state of runbook execution, start and stop jobs, view running and pending instances in real time, and review the execution history of runbook instances. The Orchestration console supersedes the OIS 6.3 Operator Console, and although the underlying technology has changed significantly, it serves the same purpose.



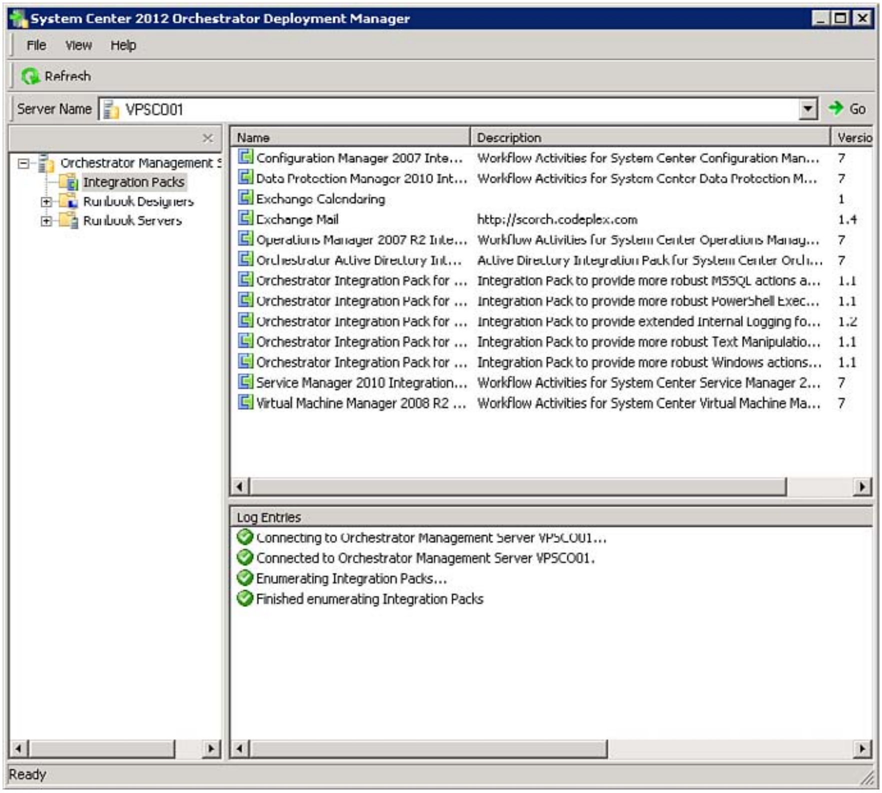
Jobs 1-50 of 66

| Status | Job ID | Runbook Name | Start Time | Last Modified Ti | Instances |
|---|---|---|---|---|---|
| ✓ | 74bfb1d7-66e0... | Start DHCP Service | 4/11/2012... | 4/11/2012... | ✓ 0 ⚠ 0 ✖ 0 |
| ✓ | 65383f63-... | SNMP Variable | 4/11/2012... | 4/11/2012... | ✓ 0 ⚠ 0 ✖ 0 |
| ✓ | 6c3452e0-... | SNMP Variable | 4/11/2012... | 4/11/2012... | ✓ 0 ⚠ 0 ✖ 0 |
| 🟥 | 15de1213-14f5-... | SNMP Alert | 4/11/2012... | 4/11/2012... | ✓ 0 ⚠ 0 ✖ 0 |
| 🟥 | af7ce07c-926b-... | SNMP Alert | 4/11/2012... | 4/11/2012... | ✓ 0 ⚠ 0 ✖ 0 |
| ✓ | 98153344-... | 1 - Create CSV | 4/10/2012... | 4/10/2012... | ✓ 0 ⚠ 0 ✖ 0 |
| ✓ | dd1cf694-096f-... | 1 - Create LOCK files | 4/10/2012... | 4/10/2012... | ✓ 0 ⚠ 0 ✖ 0 |
| 🟥 | e8be57b3-421d... | 1 - Monitor Lock File | 4/10/2012... | 4/10/2012... | ✓ 0 ⚠ 0 ✖ 0 |
| ✓ | a994f400-6e91-... | 1 - Create Test-Logfiles | 4/10/2012... | 4/10/2012... | ✓ 0 ⚠ 0 ✖ 0 |
| ✓ | fd743c07-79cc-... | 1 - Create Test-Logfiles | 4/10/2012... | 4/10/2012... | ✓ 0 ⚠ 0 ✖ 0 |

FIGURE 2.7 The Orchestration console.

### Deployment Manager

The Deployment Manager is largely unchanged from OIS 6.3 and is used to deploy runbook servers, IPs, and runbook designers. Figure 2.8 shows the Deployment Manager managing integration packs.



| Name | Description | Versio |
|---|---|---|
| Configuration Manager 2007 Inte… | Workflow Activities for System Center Configuration Man… | 7 |
| Data Protection Manager 2010 Int… | Workflow Activities for System Center Data Protection M… | 7 |
| Exchange Calendaring | | 1 |
| Exchange Mail | http://scorch.codeplex.com | 1.4 |
| Operations Manager 2007 R2 Inte… | Workflow Activities for System Center Operations Manag… | 7 |
| Orchestrator Active Directory Int… | Active Directory Integration Pack for System Center Orch… | 7 |
| Orchestrator Integration Pack for … | Integration Pack to provide more robust MSSQL actions a… | 1.1 |
| Orchestrator Integration Pack for … | Integration Pack to provide more robust PowerShell Exec… | 1.1 |
| Orchestrator Integration Pack for … | Integration Pack to provide extended Internal Logging fo… | 1.2 |
| Orchestrator Integration Pack for … | Integration Pack to provide more robust Text Manipulatio… | 1.1 |
| Orchestrator Integration Pack for … | Integration Pack to provide more robust Windows actions… | 1.1 |
| Service Manager 2010 Integration… | Workflow Activities for System Center Service Manager 2… | 7 |
| Virtual Machine Manager 2008 R2 … | Workflow Activities for System Center Virtual Machine Ma… | 7 |

```
Log Entries
Connecting to Orchestrator Management Server VPSC001...
Connected to Orchestrator Management Server VPSC001.
Enumerating Integration Packs...
Finished enumerating Integration Packs
```

FIGURE 2.8 Orchestrator Deployment Manager.

### Orchestrator Web Service

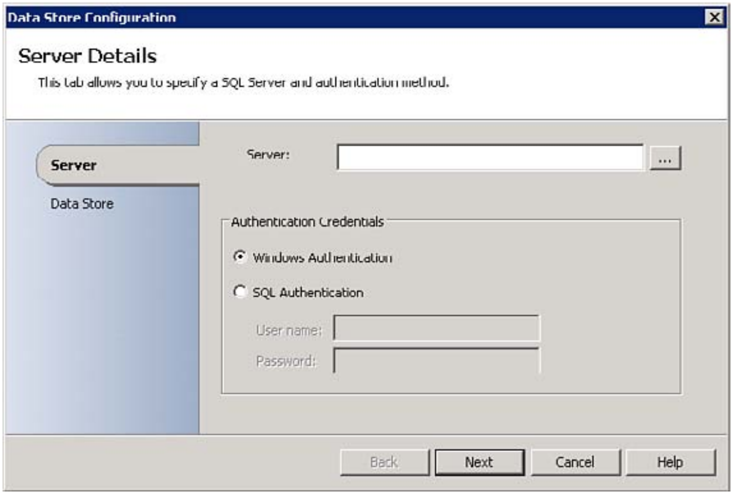
The Orchestrator web service allows for programmatic access to Orchestrator. In addition to providing access for the Orchestration console, this web service uses REST and ODATA standards to make it easier for developers to integrate their programs with Orchestrator.

### Data Store Configuration

This utility supersedes the OIS 6.3 Database Configuration Utility and is used to configure the database server and the database itself (see Figure 2.9).

## Services

Services have undergone a makeover as well. Table 2.2 lists these changes.



Data Store Configuration

Server Details

This tab allows you to specify a SQL Server and authentication method.

Server:

Authentication Credentials

● Windows Authentication

○ SQL Authentication

User name:

Password:

FIGURE 2.9 Orchestrator Data Store Configuration details.

TABLE 2.2 Services Terminology Changes

| Opalis 6.3 | Orchestrator 2012 |
|---|---|
| Opalis Remote Execution Service | Orchestrator Run Program Service |
| OpalisActionServerWatchdog | Orchestrator Runbook Server Monitor |
| OpalisActionService | Orchestrator Runbook Service |
| Opalis Management Service | Orchestrator Management Service |
| OpalisRemotingService | Orchestrator Remoting Service |

## Other Terminology Changes

Other terminology changes relate to the user interface, detailed in Table 2.3. The following sections focus on these.

TABLE 2.3 User Interface Terminology Changes

| OIS 6.3 | Orchestrator 2012 |
|---|---|
| Custom Start | Initialize Data |
| Foundation Object | Standard Activity |
| Object | Activity |
| Object Palette | Activities Pane |
| Policy | Runbook |
| Policy Folder | Runbook Folder |

| OIS 6.3 | Orchestrator 2012 |
|---|---|
| Policy Module | Job Process |
| Publish Policy Data | Published Data |
| Request | Job |
| Trigger Policy | Invoke Runbook |
| Workflow Control | Runbook Control |

### Activity

*Activity* is synonymous with *object* in OIS 6.3: It refers to the tasks dragged and dropped in the Runbook Designer to build runbooks.

### Standard Activity

Standard activities are all activities that are available in an out-of-the-box installation; they exclude activities provided by integration packs. These standard activities are sorted into different categories, based on their function. An example of these categories is Runbook Control. Chapter 7, "Runbook Basics," discusses categories for standard activities.

### Initialize Data

The **Initialize Data** activity is just a name change from the OIS **Custom Start** object, and operates in a similar way. It allows a runbook to gather user-defined input parameters. This enables runtime values to be gathered via the Orchestration console or through an interface utilizing the web service, such as the Service Manager self-service portal.

### Activities Pane

The Activities pane is the pane on the right side of the Runbook Designer that holds all the activities that can be used to build a runbook. Figure 2.10 shows the Activities pane, with some optional integration packs.

### Runbook

A runbook is synonymous with a policy in OIS 6.3: It is the collection of activities that orchestrates actions.

### Runbook Folder

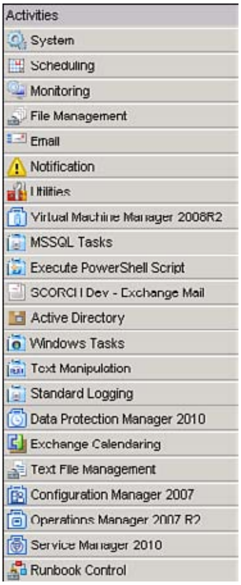
Runbook Folder replaces the legacy term Policy Folder. These folders contain one or more runbooks and are used to organize runbooks in both the Orchestration console and the Runbook Designer.

### Job

A job is a request to run a specific runbook that is waiting to be assigned to a runbook server for processing. These runbooks are assigned first come, first served.

### Job Process

A job process is the actual process that executes on the runbook server that executes an instance of a job.



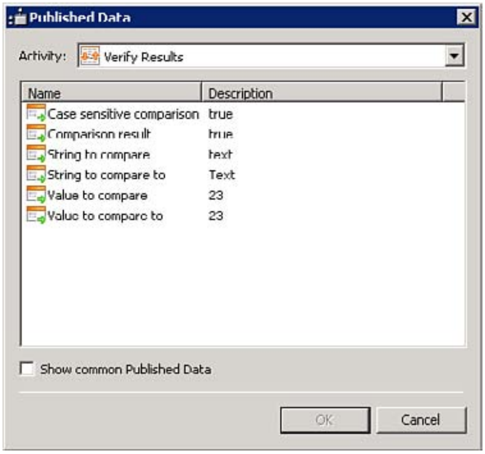
Activities

- System
- Scheduling
- Monitoring
- File Management
- Email
- Notification
- Utilities
- Virtual Machine Manager 2008R2
- MSSQL Tasks
- Execute PowerShell Script
- SCORCH Dev - Exchange Mail
- Active Directory
- Windows Tasks
- Text Manipulation
- Standard Logging
- Data Protection Manager 2010
- Exchange Calendaring
- Text File Management
- Configuration Manager 2007
- Operations Manager 2007 R2
- Service Manager 2010
- Runbook Control

FIGURE 2.10 The Activities pane in the Runbook Designer.

### Published Data

When activities run, data is collected. This includes the output of the activity, the time it ran, and whether it was successful. The information is placed in the pipeline data bus. This data can be referenced by another activity farther down the line in the runbook. Referred to as *published data*, this data was known as published policy data in OIS 6.3. Figure 2.11 shows some common published data from the **Compare Values** activity.



Published Data

Activity: Verify Results

| Name | Description |
|---|---|
| Case sensitive comparison | true |
| Comparison result | true |
| String to compare | text |
| String to compare to | Text |
| Value to compare | 23 |
| Value to compare to | 23 |

□ Show common Published Data

FIGURE 2.11 Viewing published data.

### Job

A job is a request to deploy and run a runbook on a runbook server. You can monitor jobs in the Orchestration console, previously shown in Figure 2.7. A job identifies the runbook but does not uniquely identify each specific occurrence of that runbook's execution.

Jobs can deploy a runbook to multiple runbook servers or can run multiple occurrences of the same runbook on a single runbook server. These occurrences, referred to as instances, enable you to uniquely identify each specific occurrence. For example, a System Center Operations Manager alert can trigger an Orchestrator runbook. If Operations Manager sends three alerts that are the same, the job is the request to run a runbook each time that alert is generated. The instance uniquely identifies each execution of that runbook and enables you to view data about that specific occurrence, such as the time it started and what data it generated.

### Invoke Runbook

This activity resides in the Runbook Control category and replaces the OIS legacy **Trigger Policy** object. It allows another runbook to be called from within a runbook. A related activity, **Return Data**, enables you to send back the data generated by the invoked runbook to the **Invoke Runbook** activity. This powerful pair of activities plays a big part in more complex multipart runbooks.

**CAUTION: INVOKE RUNBOOK SECURITY CREDENTIALS**

The **Invoke Runbook** activity can explicitly define security credentials that will be used by the target runbook. This is a seemingly minor change from the old **Trigger Policy** object, but the capability for an entire runbook job to be executed under specific user credentials is a significant new feature.

### Runbook Control

This activity category replaces the old Workflow Control category and contains activities that are used to control the behavior of runbooks.

## Concept Changes

Conceptually, Orchestrator has not changed much from OIS 6.3. General practices and ideas still apply, and your OIS policies largely still function in Orchestrator as runbooks. If anything, greater emphasis has been placed on the power of Orchestrator's integration with the other System Center components.

Microsoft provides updated IPs for the System Center 2012 components that leverage some of the new features and functionality in those other products. It is also worth noting that the IPs for the legacy System Center products have been updated to work with Orchestrator because the Opalis Integration Server IPs are not compatible with Orchestrator.

Previous versions required that you monitor an application for a certain event to occur in order to trigger a runbook, thus the monitor was a passive monitor. For this passive monitoring system to work reliably, the data being monitored had to be consistent enough to trigger the correct runbooks at the right time. System Center 2012 Orchestrator does not need to monitor events in external applications to trigger runbooks. Runbooks can be triggered via the web service; using integration with other applications or the System Center 2012 Service Manager component can eliminate unnecessary development efforts and issues from data inconsistencies. Chapter 6 explains this integration in more detail.

## Architecture and Feature Changes

The architecture for Orchestrator remains largely unchanged from OIS 6.3, aside from some new terminology and prerequisite changes (see Table 2.4). As Figure 2.12 shows and Chapter 3 explores further, the SQL database is still at the heart of Orchestrator. A familiar set of features operates around that SQL database.

## Prerequisite/Sizing Changes

As is typical with newly released Microsoft software, hardware and software prerequisites have been updated.

These changes should not necessarily be considered upgrade prerequisites—as stated earlier in the "OIS Migration to Orchestrator" section, no upgrade path from OIS to Orchestrator exists. Chapter 5 discusses this further.

TABLE 2.4 Single Server Prerequisite Changes

| Feature | Opalis 6.3 | Orchestrator 2012 |
|---|---|---|
| Processor | 2.1 GHz dual-core Xeon 3000 series or equivalent | 2.1 GHz dual-core Intel microprocessor or better |
| Memory | 2GB | 1GB |
| Hard Disk | 381MB | 200MB |
| Operating System Roles and Features | Windows Server 2003 SP2 or later | Windows Server 2008 R2 or Windows Server 2012 with System Center 2012 Service Pack (SP) 1, IIS, .NET Framework 3.5.1 and .NET Framework 4 |
| Database Server | SQL Server 2005 or 2008 | SQL Server 2008 R2 or SQL Server 2012 with System Center 2012 SP 1, using SQL_Latin1_General_CP1_CI_AS collation |

Apart from these relatively minor changes, the Orchestration console has been rebuilt and thus has different requirements. The old Operator Console required JavaScript on the accessing browsers and Java parts on the web server hosting the console. The new Orchestration console requires Silverlight on accessing browsers.

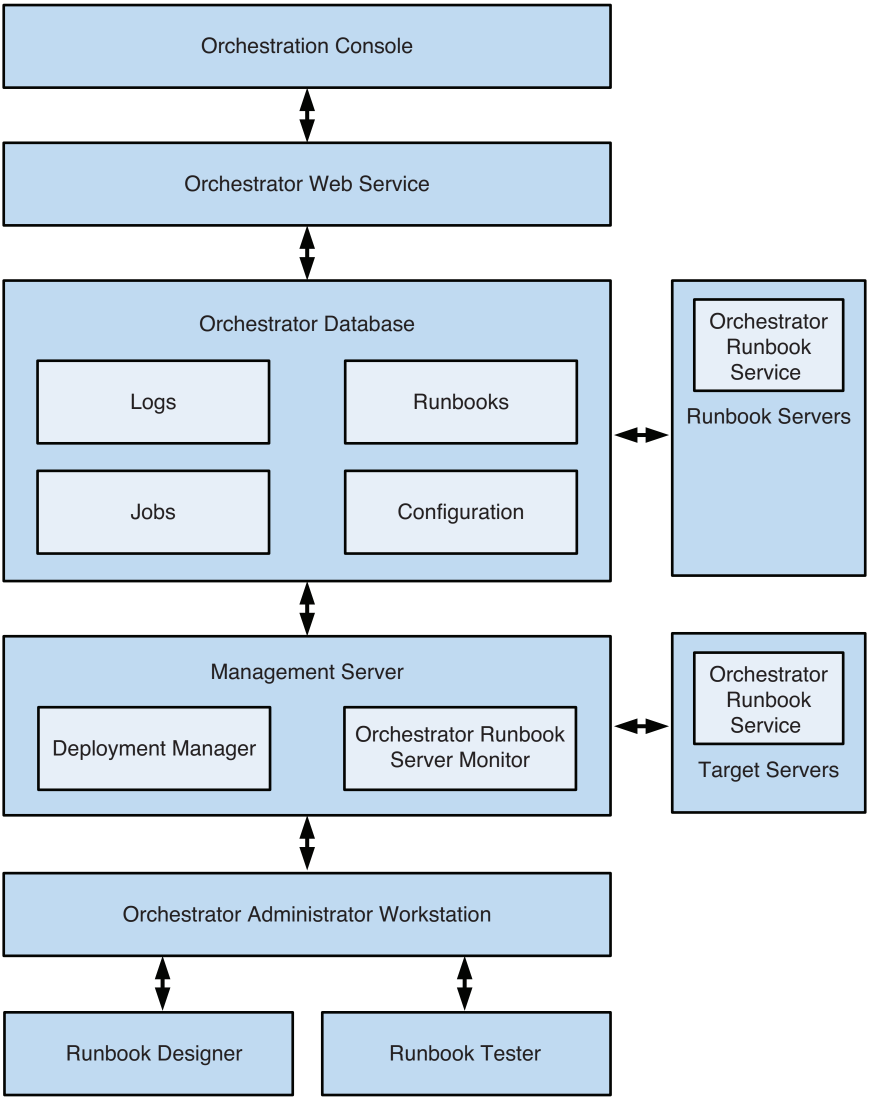


FIGURE 2.12 Architectural diagram.

Sizing and performance guidance has stayed consistent with this new release. The management server is still limited to one per environment, is needed only to connect the Runbook Designer, and does not need to be highly available. The database and runbook servers are the features required for runbooks to execute. Each runbook server is limited by default to 50 runbooks per runbook server. If you are using Service Manager with the Orchestrator connector, you will want the Orchestrator web service to be highly available as well.

## Licensing Changes

Microsoft has done a considerable amount of work to simplify the license options for System Center 2012 into an easy-to-understand processor-based licensing model. All the components of System Center 2012 have been consolidated into a single SKU, so purchasing either license offering gives you access to every component. Two editions are available; the only difference between the two is in the number of managed OSEs allowed per license (see Table 2.5).

TABLE 2.5 Licensing Changes

| License Offering | Components Included | Managed OSEs |
|---|---|---|
| System Center 2012 Datacenter Edition | App Controller<br>Configuration Manager<br>Data Protection Manager<br>Endpoint Protection<br>Operations Manager<br>Orchestrator<br>Service Manager<br>Virtual Machine Manager | Unlimited on premises, 8 in public cloud |
| System Center 2012 Standard Edition | | 2 per license on premises, 2 in public cloud |

System Center Advisor, which offers configuration monitoring cloud services for Microsoft server products, is offered at no cost to users of those products. For information on Advisor, see http://blogs.technet.com/b/momteam/archive/2013/03/06/system-center-advisor.aspx and https://www.systemcenteradvisor.com/.

# Summary

This chapter examined the evolution from OpalisRobot in 1995 to Microsoft's System Center 2012 Orchestrator. It took a close look at the differences in technology, terminology, and prerequisites. It also discussed where Orchestrator fits into System Center 2012.

The next chapter covers the Orchestrator architecture and deployment scenarios.

*This page intentionally left blank*

CHAPTER 3

# Looking Inside System Center 2012 Orchestrator

IN THIS CHAPTER

- Architectural Overview
- Server Components
- Deployment Models
- Windows Services
- Global Settings
- Integration Packs
- Runbooks
- Data Bus
- Connectors
- Consoles
- Communication

System Center 2012 Orchestrator (also referred to as SCOrch) is a new component of Microsoft System Center 2012. Microsoft acquired Opalis Software and Opalis Integration Server (OIS) in December 2009. With the System Center 2012 release, Microsoft renamed OIS to Orchestrator and updated the software to incorporate Microsoft's development standards and integrate with the most recent version of System Center.

Orchestrator enables information technology (IT) organizations to automate IT infrastructure tasks. Automation is achieved through workflows that integrate with both Microsoft and non-Microsoft products. The integration is built using activities that are delivered with integration packs.

The power of Orchestrator lies within the automated procedures and operations (known as *runbooks*) you build, and how you build them. Orchestrator brings the following key components to System Center 2012:

- **Integration:** In a data center, all the different systems need to communicate with each other. With its integration packs—software components designed around tasks targeted to a specific application—Orchestrator makes it possible for services and systems to integrate.
- **Orchestration:** The Orchestrator Runbook Designer makes it easy to drag and drop activities from different integration packs into a runbook. Activities and runbooks are configured in a graphical user interface that does not require coding.

- **Automation:** After the integration is configured and runbooks are created, these runbook can be triggered by a change request, an email, or a number of other ways. Orchestrator carries out the workflow consistently, the same way every time, and in a stable and effective way.

Orchestrator uses a different code base from other System Center components, such as Service Manager and Operations Manager, which share a common architecture and foundation. This is because of its roots from OIS, and might give you a somewhat different initial experience with Orchestrator as you start working with this component.

## Architectural Overview

The Orchestrator architecture is designed around its database. The Orchestrator database is the central feature that drives the Orchestrator environment, as all other Orchestrator features depend on data from the database. For example, all configurations are read from the database when a runbook is about to be executed, and all historical data is written to the database as the workflows complete.

Figure 3.1 presents an overview of the System Center 2012 Orchestrator architecture.

The architecture consists of these components:

- **Orchestrator database:** This is the main component and "heart" of Orchestrator; it stores all jobs, runbook, logs, and configuration information.
- **Orchestrator web service:** The Orchestrator web service, shown above the database in Figure 3.1, is a Representational State Transfer (REST)–based service that enables custom applications to connect to Orchestrator. REST defines a set of architectural principles around how to transfer information. Using the web service, other services can start and stop runbooks. The web service can also provide other services with information regarding jobs.
- **Console:** The Orchestration console is a web-based console that uses the web service to interact with Orchestrator in real time. From the console, an operator can start, stop, and view the status of jobs.
- **Runbook server(s):** Runbook servers are shown to the right side of the database in Figure 3.1. These are where an instance of a runbook runs. You can deploy multiple runbook servers in an Orchestrator environment.
- **Management server:** The figure shows only one instance of a management server because you can have only one management server in an Orchestrator environment.

  The primary purpose of the management server, shown below the database in Figure 3.1, is to provide a communications layer between the Runbook Designer and the Orchestrator database. The management server is not required to execute runbooks; these execute as long as runbook servers and the database are online. However, the Runbook Designer requires the management server. When you want to run the Runbook Designer, whether to author runbooks or start runbooks, a management system must be available. This is not the case when using the Orchestration console: Starting, stopping, and monitoring runbooks is possible with the web-based Orchestration console even if the management server is offline.

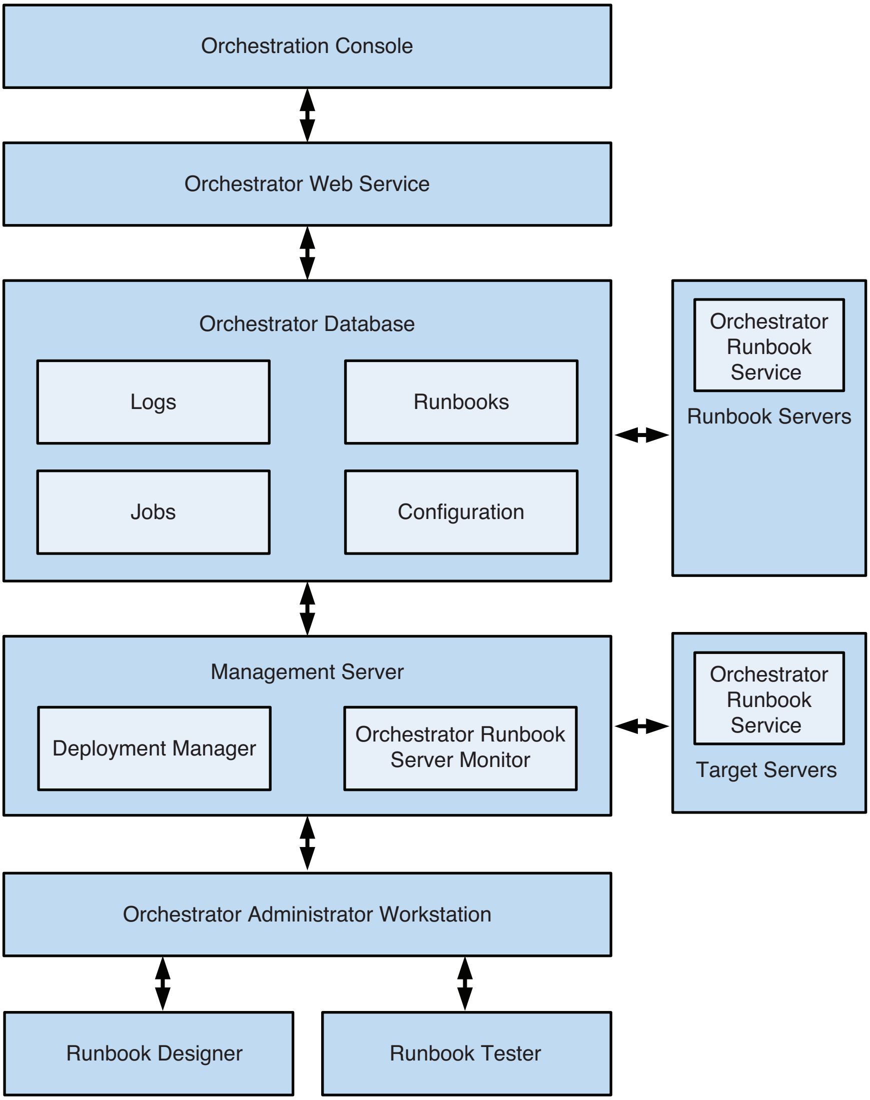


FIGURE 3.1 Architectural overview of System Center 2012 Orchestrator.

- **Runbook Service:** The runbook server uses the Orchestrator Runbook Service to run runbooks. Using this service, each runbook server communicates with the database to send status and results and to receive new jobs.
- **Runbook Designer:** This tool is used to build, modify, deploy, and manage runbooks. The Runbook Designer also contains the Runbook Tester console used for testing runbooks during development. During authoring and design, the runbooks are stored locally on the machine where the designer is running. When a runbook is

ready, you check it in, meaning that it is stored in the Orchestrator database. When you need to modify the runbook, you check it out and store a local copy that you will work with.

When you use the Runbook Designer to check a runbook into the system, the runbook is stored in the database. You can then use the Runbook Designer or Orchestration console to start the runbook. In the background, Orchestrator creates a request to run the runbook; this is referred to as a *job*. On a regular basis (every 2 seconds, by default), each runbook server checks the database for new jobs. If online, the primary runbook server for the runbook gets the job and downloads it to the local cache on the runbook server. If the primary runbook server is offline, a secondary runbook server downloads the job. When a runbook server detects the job, it logs that it is working in the job and then carries out the activities in the runbook. Status, activity results, and data are recorded in the database, enabling you to monitor the real-time and historical status of the runbook.

Deployment Manager, which is a standalone console, uses the Orchestrator Remoting Service to deploy or uninstall the Runbook Designer, runbook servers, and integration packs. The management server hosts the Orchestrator Runbook Server Monitor service, used to monitor runbook servers.

Microsoft supports virtualization of all components of System Center 2012 Orchestrator. However, even when virtualization is supported, it is not always a recommended configuration. As an example, although you could virtualize SQL Server, it is important to have a database that performs well. You often get better performance, particularly disk performance, with a physical machine that uses its own physical disks.

**NOTE: KNOW WHERE THE RUNBOOK ACTUALLY RUNS**

When you use the Runbook Tester console, the runbook executes on the local machine. This can actually cause tests to fail in some scenarios. For example, when you have installed special software on the runbook server or open firewall ports explicitly for the runbook server, the runbook will not work correctly on the local workstation. For additional information, see Chapter 11, “Security and Administration.”

## Server Components

System Center 2012 Orchestrator contains six primary features, some required and others optional. In some scenarios, you need multiple instances of some features. Runbook servers are one example.

These are the core features of System Center 2012 Orchestrator:

- Orchestrator management server
- Orchestrator runbook server
- Orchestrator database

- Orchestration console and web service (must be installed on the same machine)
- Orchestrator Runbook Designer

The following sections cover these features.

**NOTE: ORCHESTRATOR AND ACTIVE DIRECTORY**

The authors recommend that your Orchestrator servers belong to an Active Directory domain, although this it is not a requirement. You can mix components between domains and workgroups; as an example, you can have your management server and database server in one domain, with runbook servers in workgroups or other domains.

Orchestrator can work without any component belonging to a domain, but this requires extra configuration and account administration because you do not have a common account database such as Active Directory. Consider these examples:

- If the runbook servers cannot connect with a domain account to the SQL Server hosting the Orchestrator database, you must configure the SQL Server to use SQL authentication. Microsoft does not recommend SQL authentication because it is less secure than Windows authentication.
- Security roles control who can access and modify runbooks in the Runbook Designer. These roles are more complex to maintain without domain accounts. If you use local (nondomain) accounts, you must administer security settings locally on the Orchestrator management server. If you use Active Directory security groups, you can simply add a user to a security group to grant him or her permissions in Orchestrator.

## Management Server

The management server provides a layer between the Runbook Designer and the Orchestrator database. Only one management server can exist in an Orchestrator environment. Orchestrator does not provide the means to have a fault-tolerant management server because this role cannot be clustered.

The management server is not required to execute runbooks. Runbook servers communicate directly with the database to obtain jobs to execute and to report results. The management server is needed only when connecting to the Runbook Designer console or Deployment Manager console. If you need to recover the management server in a disaster scenario, you can reinstall the role and point it to the Orchestrator database.

The Orchestrator Deployment Manager is installed on the management server by default. Deployment Manager is used to deploy new runbook servers, integration packs, and runbook designer consoles. If the management server is offline, you cannot deploy additional runbook designer consoles. However, you can manually install runbook servers and integration packs when Deployment Manager is offline.

**TIP: MANUAL INSTALLATIONS**

Consider some additional information about manual installations:

- Information on deploying integration packs manually is located on the TechNet Wiki at http://social.technet.microsoft.com/wiki/contents/articles/how-to-install-opalis-integration-server-action-server-without-using-deployment-manager.aspx. (Although this refers to an Opalis action server, the process is very similar to that in Orchestrator.)
- Additional information regarding runbook server deployments is available at http://technet.microsoft.com/en-us/library/hh420386.aspx.
- Information about runbook designer deployments is located at http://technet.microsoft.com/en-us/library/hh420343.aspx.

## Runbook Server

Runbooks are executed on runbook servers. A runbook server communicates directly with the Orchestrator database to determine new jobs to execute and to report results. You can deploy multiple runbook servers, depending on constraints such as network requirements and number of runbooks that need to run at the same time.

When a request to start a runbook comes in (manually by an operator or administrator, or through a scheduled activity or other automated mechanism), a job is created in the Orchestrator database. When a runbook server finds the job, it marks it as in progress, copies the runbook locally, and begins executing the runbook. Status, results from activities, and data are inserted into the Orchestrator database. You can view this data in the Orchestrator runbook designer consoles. Results can also be read from the Orchestrator web service. The Orchestration console connects through the Orchestrator web service to the Orchestrator database.

## Orchestrator Database

The Orchestrator database is a Microsoft SQL Server database. It is the central component of Orchestrator. The database stores runbooks, configuration, logs, and the status of running runbooks. Without the database, your Orchestrator environment is not functional. The authors recommend that you deploy the Orchestrator database on a well-performing and fault-tolerant SQL Server.

## Web Service

Using the Orchestrator web service, custom applications can connect to Orchestrator to start and stop runbooks. They can also get information regarding running runbooks. The Orchestration console uses the web service to connect to the Orchestrator environment. The web service communicates directly with the database.

## Orchestration Console

The Orchestration console is a web-based console you can use to start and stop runbooks. An important distinction between the Orchestration console and the Runbook Designer is