



Microsoft System Center

Optimizing Service Manager

Thomas Ellermann • Kathleen Wilson • Karsten Nielsen • John Clark
Mitch Tulloch, Series Editor

Visit us today at



microsoftpressstore.com

- **Hundreds of titles available** – Books, eBooks, and online resources from industry experts
- **Free U.S. shipping**
- **eBooks in multiple formats** – Read on your computer, tablet, mobile device, or e-reader
- **Print & eBook Best Value Packs**
- **eBook Deal of the Week** – Save up to 60% on featured titles
- **Newsletter and special offers** – Be the first to hear about new releases, specials, and more
- **Register your book** – Get additional benefits



Hear about it first.

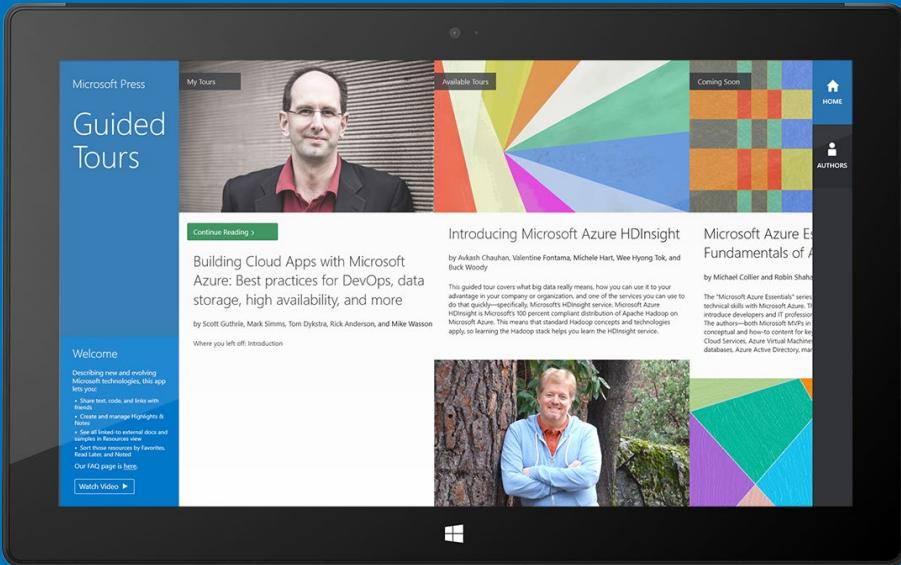


Get the latest news from Microsoft Press sent to your inbox.

- New and upcoming books
- Special offers
- Free eBooks
- How-to articles

Sign up today at MicrosoftPressStore.com/Newsletters

Wait, there's more...



Find more great content and resources in the **Microsoft Press Guided Tours** app.



The [Microsoft Press Guided Tours](#) app provides insightful tours by Microsoft Press authors of new and evolving Microsoft technologies.

- Share text, code, illustrations, videos, and links with peers and friends
- Create and manage highlights and notes
- View resources and download code samples
- Tag resources as favorites or to read later
- Watch explanatory videos
- Copy complete code listings and scripts





From technical overviews to drilldowns on special topics, get *free* ebooks from Microsoft Press at:

www.microsoftvirtualacademy.com/ebooks

Download your free ebooks in PDF, EPUB, and/or Mobi for Kindle formats.

Look for other great resources at Microsoft Virtual Academy, where you can learn new skills and help advance your career with free Microsoft training delivered by experts.

Microsoft Press

PUBLISHED BY
Microsoft Press
A Division of Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399
Copyright © 2013 by Microsoft Corporation (All)
All rights reserved. No part of the contents of this book may be reproduced or transmitted in
any form or by any means without the written permission of the publisher.
Library of Congress Control Number: 2013956656
ISBN: 978-0-7356-8312-9

Printed and bound in the United States of America.

First Printing

Microsoft Press books are available through booksellers and distributors worldwide. If you
need support related to this book, email Microsoft Press Book Support at
mspininput@microsoft.com. Please tell us what you think of this book at
<http://www.microsoft.com/learning/booksurvey>.

Microsoft and the trademarks listed at <http://www.microsoft.com/en-us/legal/intellectualproperty/Trademarks/EN-US.aspx> are trademarks of the Microsoft group of
companies. All other marks are property of their respective owners.

The example companies, organizations, products, domain names, email addresses, logos,
people, places, and events depicted herein are fictitious. No association with any real company,
organization, product, domain name, email address, logo, person, place, or event is intended
or should be inferred.

This book expresses the author's views and opinions. The information contained in this book is
provided without any express, statutory, or implied warranties. Neither the authors, Microsoft
Corporation, nor its resellers, or distributors will be held liable for any damages caused or
alleged to be caused either directly or indirectly by this book.

Acquisitions Editor: Anne Hamilton
Developmental Editor: Karen Szall
Project Editor: Karen Szall
Editorial Production: Megan Smith-Creed
Copyeditor: Megan Smith-Creed
Cover Illustration: Twist Creative, Seattle
Cover Design: Microsoft Press Brand Team

Contents

<i>Foreword</i>	<i>vii</i>
<i>Introduction</i>	<i>ix</i>
Chapter 1 Business reasons to choose Service Manager	1
Introduction	1
Integration story.....	3
Active Directory connector	3
Configuration Manager.....	3
Operations Manager.....	3
Orchestrator	4
Exchange connector.....	4
Reporting.....	4
Chapter 2 Deployment costs and non-IT usage	7
Licensing	7
Deployment costs	9
Envision	9
Plan	10
Build.....	11
Stabilize	11
Deploy	12
Operational costs	12
Non-IT usage	12
Request a new credit card	13
Request access to an invoicing system.....	14
Using Service Manager for customer service.....	14

What do you think of this book? We want to hear from you!

Microsoft is interested in hearing your feedback so we can continually improve our books and learning resources for you. To participate in a brief online survey, please visit:

microsoft.com/learning/booksurvey

Chapter 3	How to plan for Service Manager	17
	It's not just the technology	17
	People.....	18
	Process.....	19
	Product.....	21
	Implementation roles	22
Chapter 4	How to prepare for a Service Manager installation	25
	Technology considerations	25
	Active Directory	25
	Operations Manager	26
	Configuration Manager.....	28
	Virtual Machine Manager	28
	Process considerations.....	29
	Incident management.....	29
	Problem management.....	30
	Change management.....	31
	Release management.....	32
	Service request management	33
Chapter 5	Management packs	35
	Management pack general guidance.....	35
	Management pack naming guidance	35
	Bundling modifications.....	36
	Naming and bundling views and view folders.....	37
	Naming and bundling templates	38
	Naming and bundling service offerings and request offerings	38
	Naming and bundling groups and queues	38
	Naming and bundling console tasks.....	38
	Naming and bundling notification templates and subscriptions.....	39
	Sealing management packs.....	39
	Updating a sealed management pack	40
	Versioning management packs.....	41
	Backing up management packs.....	42
	Renaming management pack filenames.....	42

Chapter 6	Optimizing the Service Manager environment	45
Service Manager management server.....	46	
Service Manager console.....	46	
Service Manager databases	47	
SQL Server editions.....	49	
Workflows	50	
Service Manager Self-Service Portal.....	51	
Connectors.....	52	
General considerations	53	
Active Directory connector	53	
Operations Manager connector.....	56	
Configuration Manager connector	56	
Orchestrator connector	57	
Chapter 7	Service Manager configuration and customization	59
Configurations.....	60	
Incident and service request support groups and assignment	60	
Incident categorization.....	60	
Customizations.....	62	
Work item custom labeled fields	63	
Notification Boolean.....	66	
Assignment notification for all work items.....	67	
More information needed and possible problem.....	68	
Location on form.....	69	
On Behalf Of on form.....	70	
VIP incidents and requests	71	
Submitting change requests from the Self-Service Portal.....	72	
Change phase in change views	75	
Additional resources for configuration and customization	77	
Customization risk areas	77	
Orchestrator versus Authoring Tool for workflows	78	

What do you think of this book? We want to hear from you!

Microsoft is interested in hearing your feedback so we can continually improve our books and learning resources for you. To participate in a brief online survey, please visit:

microsoft.com/learning/booksurvey

This page left intentionally blank

Foreword

A high percentage of Microsoft System Center 2012 Service Manager projects don't deliver on their promises as they should. While each Service Manager implementation has its own unique challenges, all the successful projects share certain common attributes and experiences. In this book, the authors, Thomas Ellermann, John Clark, Kathleen Wilson, and Karsten Nielsen, who collectively represent close to 60 years of IT consulting experience, express this sentiment and provide a blueprint to help deliver successful Service Manager implementations. This is an undertaking of immense value to the community, and I am honored to be writing this foreword and to recommend the book.

This book is not just for new Service Manager projects. The information presented here benefits existing implementations that are in dire need of optimization. This book is also not a substitute for obtaining detailed knowledge on Service Manager technical information or attending training sessions on Service Manager.

This book is about the organizing principle of Service Manager projects and the various roles in the organization that impact the project. In my experience talking to customers, choosing an ITSM solution today is one of the most difficult challenges facing an IT organization. There are close to 350 vendors claiming this space. Coupled with build-your-own alternatives and an ever decreasing IT budget, a host of certifying bodies, analyst recommendations, and the reality of the disrupting effect of the cloud technologies, selecting an ITSM solution becomes a daunting task, even for the experts. The authors of this book understand this complexity. They have taken the lessons from successful Service Manager implementations and have created a framework that can be leveraged by various stakeholders in an organization to move the needle toward a service oriented delivery model.

Any product so pivotal to changes in business process is bound to have its share of shortcomings. Service Manager is no exception, and the authors explicitly call on the dependencies and shortcomings of Service Manager, making it easier for you to make informed choices. The authors further call on you to challenge your assumptions and pave an improved path to efficiencies that come with automation and standardization.

This book will offer you at least three benefits: You will learn about the capabilities of Service Manager and how it can help you transform service delivery in the modern service-centric business. You will learn how to plan and prepare a Service Manager project. Lastly, you will learn to optimize your current implementation, know about the partner solutions in this space, and improve the productivity of your offerings.

I enjoyed the book and found it valuable. I hope you will, too.

Ranganathan Srikanth

Principal Program Manager, Windows Server System Center (WSSC) at Microsoft Corporation

This page left intentionally blank

Introduction

Welcome to *Microsoft System Center: Optimizing Service Manager*. We (the authors) all work with systems management at Microsoft and believe that the Microsoft System Center suite is one of the most integrated suites on the market for this purpose.

Microsoft System Center 2012 Service Manager is the only product that can integrate across most of the System Center suite and Active Directory. Service Manager is a fast and reliable product that can create and maintain a dynamic service management database to enable interaction across the organization, both inside and outside the IT department, making it a very compelling product to many organizations.

Over the last several years, more and more customers have implemented Service Manager, either independently or via Microsoft or a partner. Sometimes the project and product implementation are not as successful as they should be. Our objectives with this book are to provide you with a framework for planning and delivering a successful Service Manager project and to share some of our experiences and best practices when it comes to optimizing and maintaining your Service Manager environment.

This book is written with three different roles in mind: business and technical decision makers; IT architects; and Service Manager administrators. You can either read this book in its entirety from A to Z, or you can follow one of the learning paths below depending on your role:

- Business and technical decision makers:
 - Chapter 1 Business reasons to choose Service Manager
 - Chapter 2 Deployment costs and non-IT usage
- IT architects
 - Chapter 3 How to plan for Service Manager
 - Chapter 4 How to prepare for a Service Manager installation
- Service Manager administrators
 - Chapter 5 Management packs
 - Chapter 6 Optimizing the Service Manager environment
 - Chapter 7 Service Manager configuration and customization

About the companion content

The companion content for this book can be downloaded from the following page:

<http://aka.ms/SCserviceMgr/files>

Acknowledgments

We would like to thank the following people who helped review the content of this book for technical accuracy:

- Rob van der Burg, Business Program Manager, Microsoft Netherlands
- Jon Sabberton, Architect, Microsoft Gulf, United Arab Emirates
- Murat Erentürk, Senior Architect, Microsoft MEA HQ

Errata & book support

We've made every effort to ensure the accuracy of this content and its companion content. Any errors that have been reported since this content was published are listed on our Microsoft Press site:

<http://aka.ms/SCserviceMgr/errata>

If you find an error that is not already listed, you can report it to us through the same page.

If you need additional support, email Microsoft Press Book Support at
mspinput@microsoft.com.

Please note that product support for Microsoft software is not offered through the addresses above.

We want to hear from you

At Microsoft Press, your satisfaction is our top priority, and your feedback our most valuable asset. Please tell us what you think of this book at:

<http://aka.ms/tellpress>

The survey is short, and we read every one of your comments and ideas. Thanks in advance for your input!

Stay in touch

Let's keep the conversation going! We're on Twitter: *<http://twitter.com/MicrosoftPress>*.

Business reasons to choose Service Manager

Choosing an IT Service Management (ITSM) solution today is one of the more difficult challenges an IT organization faces. This chapter describes some of the business reasons for selecting Microsoft System Center 2012 Service Manager as your organization's ITSM platform.

Introduction

With well over 400 competitors in the marketplace, multiple certifying bodies, and the over-abundance of industry analysis schemes, it's ironic that customers quite often still struggle to achieve outcomes they intended when they invested in an ITIL-certified, industry analyst-identified "leading" ITSM solution.

Some of the issues are these:

- Many ITSM tools today either do not take the customer past basic technical ticketing, or the customer doesn't implement the tool for true, business-process-aligned service management.
- Customers put governance of their ITSM tools into the same technical or lifecycle siloes that already are a cause of dysfunction in their organization. ITSM tools need to be managed, maintained, and optimized as the organization improves and matures their service delivery.
- True service management is seen as insincere when services are not managed, dependencies are not mapped and managed, and knowledge is not shared across all the white space of the IT organization.
- Organizations take a "best of breed" approach, placing limits on the capabilities of all of the solutions selected.

According to Gartner Group (*Top 10 IT Service Management Next Practices, G00237446, Published: 12 June 2013*), process continues to be the least mature dimension of an IT organization. Organizations that are not leveraging ITIL or COBIT correctly will remain at a very low level in the Infrastructure and Operations

Maturity (ITSIO) scale. Any possibility of moving from a technology-centric cost center to a service-centric business model is just not happening in many organizations worldwide.

Customers looking at implementing a solution that mimics the traditional ticketing systems like BMC Remedy, HP Service Manager, or CA Service Desk are wise to recognize that System Center Service Manager was not intended to simply replace many of the manual workloads and practices legacy ITSM solutions implemented, but to eliminate or reduce them through standardization, integration, and extensive automation.

The first and foremost value proposition of Service Manager begins with the fact that it is included with Enterprise CALs of System Center 2012 (more information of license types can be found at <http://www.microsoft.com/calsuites/en/us/default.aspx>). Many customers do not even realize they already own Service Manager because of their investment in Microsoft System Center solutions. It is important to check with your Microsoft Account Manager to be sure you are properly licensed. So the cost of entry has already been paid, usually requiring some implementation services and training to make the most of Service Manager.

With that said, one might conclude that a solution with half the capabilities of Service Manager is "worth it" due to no additional software costs. However, like any Service Management solution, Service Manager has strengths and weaknesses, and many of its strengths put it into a class of its own among ITSM competitors. It really comes down to whether or not the weaknesses are something a customer can work with or around.

Service Manager is different from most ITSM solutions on the market today for a number of reasons:

- Service Manager would never be proposed by Microsoft as a stand-alone solution without being implemented alongside System Center 2012. If you are implementing Service Manager by itself, stop and reconsider the value that comes from implementing System Center holistically.
- Service Manager is highly dependent on the data available to it from Active Directory, Configuration Manager, Operations Manager, Virtual Machine Manager, and even Orchestrator and Exchange, to operate properly. Without these built-in integrations, and easily implemented integrations, the data value of Service Manager is limited.
- Service Manager's Data Warehouse is unique by industry standards in that it houses a built-in OLAP-based data warehouse, which allows users unfettered access to analysis data. Very few solutions on the market have this capability out of box, and even if they do, you still need to purchase a reporting module.

ITM organizations need to take a new approach to service management, one that addresses one or more of the core tenants of IT value: cost reduction, business growth or transformation, quality improvement, and compliance. Managing technology as a service has proven to be the way to understand and communicate IT value. System Center allows you to manage from both a technical perspective and a service perspective.

Integration story

As highlighted earlier, the integration of Service Manager with the rest of the System Center suite and with Active Directory makes it one of the most mature offerings in the ITSM market. The following examples highlight some of the value areas that result from the built-in integrations of System Center 2012.

Active Directory connector

The Active Directory connector is a one-way connector between Service Manager and Active Directory Domain Services (AD DS). The Active Directory connector is able to import users and groups to the CMDB, as well as printer and computer objects. Leveraging the Active Directory connector for people and group management, Service Manager provides value from the following:

- Roles in Service Manager can be assigned to security groups from Active Directory, reducing the amount of time necessary to manage users and rights in Service Manager.
- Getting group membership information from Active Directory versus managing groups separately in various ITSM solutions reduces the amount of time necessary to manage groups of users in Service Manager.
- Active Directory groups are leveraged for personalization of the Self-Service Portal, which allows targeted services to specific stakeholders inside and outside of the IT organization.

Configuration Manager

The Configuration Manager connector is a one-way connector between Service Manager and Configuration Manager. Configuration Manager provides rich, in-depth information about computers and servers that are managed by it. The connector will import and maintain information about installed software, installed patches, and which user is the primary user of a computer. Leveraging the Configuration Manager connector for computer and system management, Service Manager provides value from the following:

- Rich, in-depth data about computers, software, devices, and other information about managed objects
- Ability to audit client systems

Operations Manager

The Operations Manager Configuration Item connector is a one-way connector between Service Manager and Operations Manager while the Operations Manager Alert connector is a two-way connector. These two connectors enable not only the import of monitored

configuration items into the CMDB, but also the ability to create alerts as incidents in Service Manager, which enables Service Manager to be used as an infrastructure management tool as well.

Operations Manager has a powerful capability called Distributed Applications that monitors the health of a service based on the health of all components that are a part of the service. Distributed Applications can then be imported automatically into the CMDB as a Business Service, where additional service properties can be managed, including customers, key contacts, and so on. Leveraging the Operations Manager connectors for computer and system management, Service Manager provides value from the service health views of monitored objects.

Orchestrator

The Orchestrator connector is a two-way connector between Service Manager and Orchestrator. The Orchestrator connector provides a seamless, web service-based interface between Service Manager and Orchestrator that allows Orchestrator runbooks to be embedded within work items in Service Manager. Runbooks can then be started, for example from a Service Request, to perform various automated activities. Leveraging the Orchestrator connector for computer and system management, Service Manager provides value from quick implementation of Runbook Automation with Service Manager.

Exchange connector

The Exchange connector is a one-way connector that provides a seamless, web service-based interface between Service Manager and Microsoft Exchange, which allows emails to be used for the management of the lifecycle of work items. Leveraging the Exchange connector provides value from the following:

- Enables rapid updates to the work items even without access to the console or Self-Service Portal
- Provides a familiar interface for users when interacting with Service Manager

Reporting

Along with the inherent integration supported by Service Manager, the reporting and dashboard capabilities included in the solution put it far above virtually any ITSM solution on the market. By and large, good reporting capabilities are the missing ingredient in most other ITSM market solutions. Often the solution proposed for third-party ITSM solutions is Crystal Reports or some form of proprietary reporting.

Service Manager takes advantage of Microsoft business intelligence solutions incorporated into SQL Server. There are two options for reporting from Service Manager:

- **Reporting Services** Through Microsoft SQL Server Reporting Services (SSRS), "transactional" reports are available through both the Service Manager console as well as the SQL Server Reporting Services web portal. These reports are often tabular and follow the relational data model of Service Manager.
- **Analysis Services** Through the use of Data Cubes, together with data from Configuration Manager and Operations Manager, Service Manager supports the use of industry standard OLAP cubes that can be queried by Microsoft Excel, Microsoft SharePoint, or any third-party tool that can read SQL Server OLAP cubes. This allows for deeper analysis of multi-dimensional data that results from the relationship-based data model within Service Manager.

See Figures 1-1, 1-2, and 1-3 for some examples of the reporting capabilities of Service Manager using the Service Manager console, Microsoft Excel, and the SharePoint Dashboard.

By leveraging these reporting and SharePoint dashboard capabilities, Service Manager customers can realize the elusive value that is always promised by other solution makers, that leverages modern business intelligence technologies through Microsoft SQL Server.

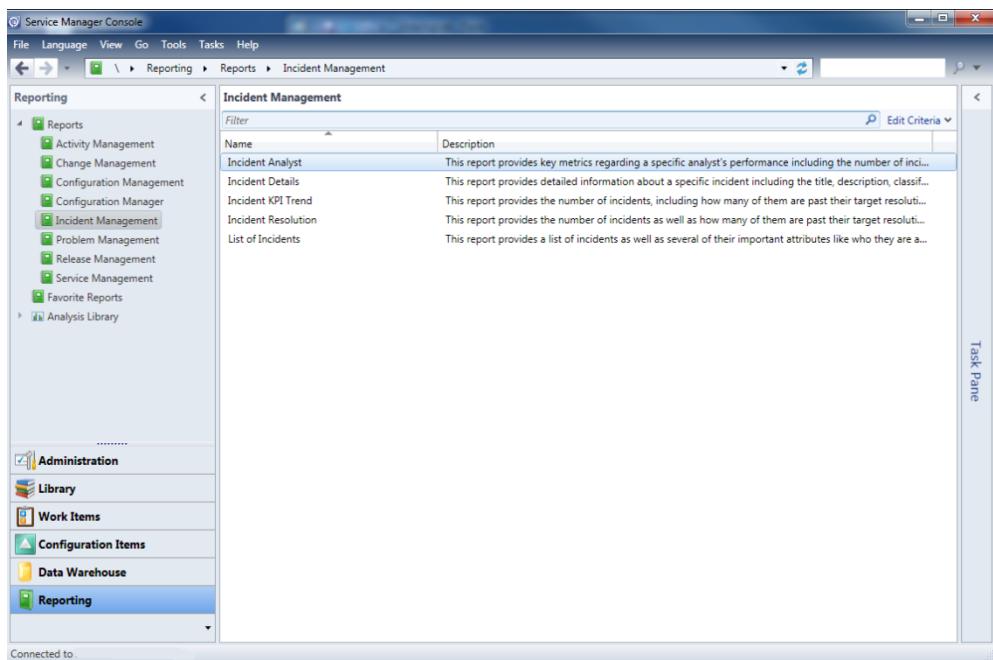


FIGURE 1-1 Reporting in the Service Manager console.

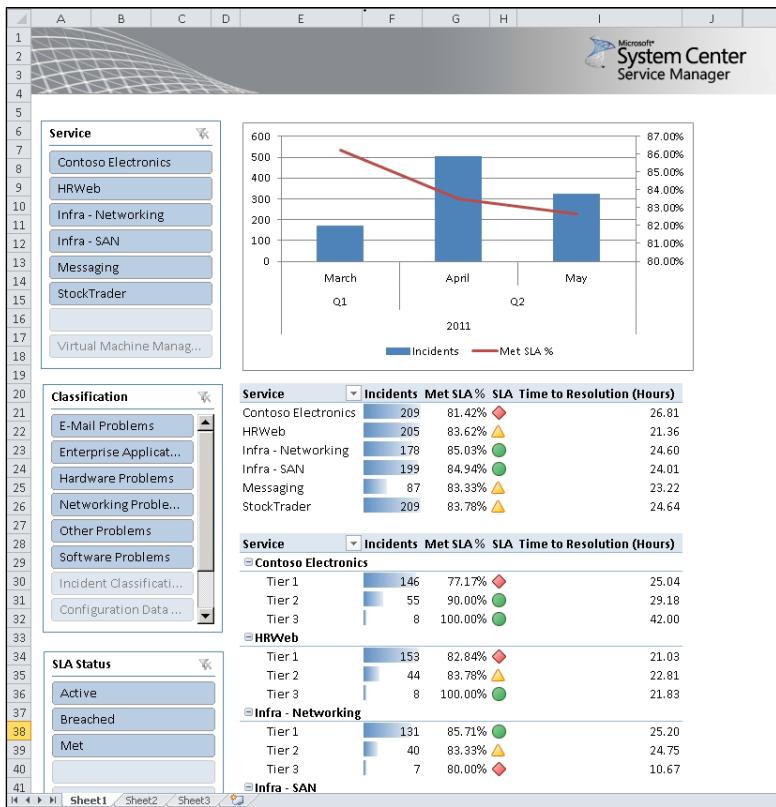


FIGURE 1-2 Viewing reports in Microsoft Excel.



FIGURE 1-3 Viewing reports in SharePoint.

Deployment costs and non-IT usage

The cost of deploying Microsoft System Center 2012 Service Manager must be understood prior to implementation. While many customers already own Service Manager as part of their Microsoft volume licensing agreement, some funding is necessary to ensure that the proper scope, planning, and resources needed for deployment are properly accounted for and are not merely an afterthought. This chapter summarizes the deployment costs involved with implementing Service Manager. The chapter also describes some scenarios where customers have used Service Manager to manage requests for groups outside of IT.

Licensing

Service Manager is part of the System Center suite of products, and the components included in this suite share a common licensing model. You cannot buy licenses for individual System Center products. In the past you could purchase individual product licenses, but two System Center products would be as expensive as the suite license. With the current System Center suite license model, you get the whole stack. With access to the full suite of products, you can try the different System Center products to realize the value of the entire suite.

IMPORTANT The System Center licensing model described here is valid as of October 2013 and may be changed at any time.

The following products are included in the System Center suite license:

- Configuration Manager
- Service Manager
- Virtual Machine Manager
- Operations Manager
- Data Protection Manager
- Orchestrator
- App Controller
- Endpoint Protection

There are two different types of System Center licenses: a Standard edition and a Datacenter edition. The Standard and Datacenter editions of the System Center 2012 server management licenses differ based only on the number of operating system environments (OSEs) that may be managed. System Center 2012 Standard licenses allow customers to manage two OSEs on premises or two OSEs in a public cloud environment. System Center 2012 Datacenter licenses cover an unlimited number of OSEs for an on-premise environment or eight OSEs in a public cloud environment.

The System Center license is included in the SQL Server license (Standard Edition), but SQL Server Enterprise is recommended for the data warehouse. Many customers will also need client licenses for several System Center products that also require client integration. You can purchase these licenses through specific System Center client licenses or through the normal Microsoft client access licenses (CALs). For the System Center client license, there are three possibilities, as outlined in Table 2-1. If you own the Core CAL Suite or the Enterprise CAL Suite, you also have access to the System Center licenses, as shown in Table 2-2.

TABLE 2-1 Summary of System Center client licensing

LICENSE MODEL	PRODUCTS INCLUDED
System Center 2012 Configuration Manager Client ML	Configuration Manager Virtual Machine Manager
System Center 2012 Endpoint Protection Client ML	Endpoint Protection
System Center 2012 Client Management Suite Client ML	Service Manager Operations Manager Data Protection Manager Orchestrator

TABLE 2-2 System Center client licenses included in Core or Enterprise CAL Suite

LICENSE MODEL	CAL SUITE
System Center 2012 Configuration Manager Client ML	Included in Core CAL Suite Included in Enterprise CAL Suite
System Center 2012 Endpoint Protection Client ML	Included in Core CAL Suite Included in Enterprise CAL Suite
System Center 2012 Client Management Suite Client ML	Included in Enterprise CAL Suite

Deployment costs

The cost to deploy Service Manager will depend on the management packs and functionality you require. Deploying Service Manager involves not only deploying new technology but also consideration of people and processes such as:

- Will you be using in-house or external resources for your deployment?
- Do you have other System Center products in place?
- Does your environment satisfy the prerequisites for deploying Service Manager?
- Are your processes well-defined?
- What processes will be implemented during each phase of your deployment?
- How advanced are the process requirements?

The following sections outline the approach used by Microsoft Consulting Services (MCS) when deploying Service Manager. This outline can be helpful in determining your potential deployment costs.

Envision

The key activity in this phase is the envisioning workshop. The goal of this workshop is to define a shared vision for the Service Manager project team and the groups that will be using Service Manager. The outcome of the envisioning workshop is a Vision Scope document that clearly articulates the "why" and "what" the organization expects to achieve from implementing Service Manager. An important part of the Vision Scope document is to define the project's scope in such detail that it is not questioned later. At this stage of the deployment process you should avoid discussing how Service Manager will perform a particular task. This "how" aspect of Service Manager will occur during the build phase which is described later in this chapter.

Setting the vision and scope at the onset of the deployment helps ensure that all involved parties of the organization are on the same page concerning what they expect implementing Service Manager will help them achieve. The envisioning workshop is essential for the project because its output is the foundation for all decisions made during the project. An example of a vision/scope statement might be:

"Contoso has a vision of being able to manage its Microsoft environment and critical applications in a more mature and efficient way by combining their IT-processes and the System Center platform into one solution to initiate their IT Service Management vision. This solution will increase the availability and reliability of the environment and allow Contoso to better manage and extend their Service Management processes."

The envisioning workshop should also cover knowledge transfer so that the project team is aware they must actively participate in the deployment. For more information on who should participate in the envisioning workshop, see Chapter 3, "How to plan for Service Manager."

Plan

Two key planning activities are developing the project schedule and planning the envisioning workshops. As far as project scheduling is concerned, it's important to plan for the Service Manager implementation by creating a project plan for the entire project including all milestones and resources needed for deployment.

The first thing you need to do is plan the envisioning workshops. You will typically need to schedule multiple workshops and keep the participants under 15 people. The workshops are used to collect information concerning the current state of the people, processes, and technology in the organization. Table 2-3 is a list of recommended workshops, but the processes and functionality of Service Manager you will be implementing will determine which of these you do. For more information on what to discuss during these workshops, see Chapter 4, "How to prepare for a Service Manager installation."

TABLE 2-3 List of envisioning workshops

WORKSHOP NAME	OBJECTIVES OF WORKSHOP
Reporting Requirements	Starting with the reporting requirements is the best way to understand what management needs to report on from Service Manager. This will help you identify what you need to track and measure in Service Manager.
Configuration Management Envisioning	During this workshop, discussion centers around what connectors will be used to import data into Service Manager.
Incident Management	The incident management process owners will be in attendance to share the process and discuss how Incident management works. Ensure that you have help desk, tier 2, and management participation.
Service Request/Service Catalog	During this workshop, the service request process will be discussed. The help desk will be key to the discussion as well as other groups that are involved in the provisioning of new services. Note that if self service is planned with Service Manager, you will also need to discuss the current service catalog since the portal will need to imitate this. Because customers will be navigating the portal, you need to make sure the portal uses language that the customers (end users) are familiar with.
Problem Management	The problem management process owners will be in attendance to share the process and discuss how problem management works. Note that many of the problem analysts will work in tier 2 and on incident management

Change Management	The change management process owners will be in attendance to share the process and discuss how change management works. Change initiators will also have to be in attendance since they have hands-on knowledge of the process
Release Management	The release manager will need to participate in this workshop as well as the project managers who implement IT projects. A word of caution: If change management has not been implemented in Service Manager, you should hold off implementing release management until the organization is familiar with change management and service requests.

Once the workshops have been completed, the results should be documented in a Functional Specification document—one or more documents that contain a detailed description of the full solution, including design and configuration of the tool, workflows, and processes. Functional Specification should be based on the Vision Scope document. It is the foundation for a Service Manager implementation and should be treated as the blueprint for implementation and ongoing maintenance of Service Manager.

Build

During the build phase, the physical implementation of the processes happens in Service Manager. This is often referred to as the "how" stage of Service Manager deployment. Examples of customizations performed during this phase might include modifications to the drop down lists, creation of notifications, creation of workflows and templates, and so on.

Once completed, a series of demos and process-based walkthroughs should take place with the process owners to ensure that Service Manager has been modified as per the process requirements. Process owners should test the build to ensure that Service Manager works as intended before the next phase begins.

Stabilize

Run pilots during the stabilization phase and track results and issues to be addressed prior to production implementation. During this phase, provide training to everyone in the organization including the customers who will be using the portal to log requests and incidents. You should train all of your IT staff to use Service Manager to avoid a flood of complaints when they actually start using it. During this phase you should also update your documentation, including your Functional Specification document and Operations Guide.

The final step of this phase is the operational review and sign-off of the Service Manager solution. This is where IT accepts the Service Manager solution as deployed and ready to move to a production release.

Deploy

During this phase, Service Manager is deployed in production. Make sure there is proper support coverage for the first week; there will be plenty of questions on how to use Service Manager. What it costs to manage Service Manager will vary, but the maintenance of the platform (the technical layer) will require about 4 hours of work per week to maintain, monitor, and apply technical updates to Service Manager.

Operational costs

Many organizations think that once they have deployed Service Manager, they are done, but this is not the case. The operational costs of managing and maintaining Service Manager include not only the ongoing maintenance of the platform but also any process improvements, new features, or functionalities you decide that you need to add to Service Manager. For example, consider management of drop-down lists, adding new workflows, or meeting additional reporting requirements.

One area that does not get as much attention as it should is ongoing improvement and management of process management packs from a business and process perspective. What this means in ITIL terms is Continual Service Improvement (CSI). Each process management pack should be owned and maintained by the process owner. For example, the incident management process owner is responsible for ensuring that any changes to the Incident Management Process management pack are aligned with evolving incident management processes. Each process owner will need to ensure that the drop-down lists, workflows, notifications, and reports not only support the current state of the process but improve service support and delivery.

Depending on where your organization is with its ITSM processes and whether or not you have embraced CSI, the ongoing operations of Service Manager may vary due to what you are trying to achieve from a process perspective.

Non-IT usage

It's not only IT that assists users within an organization; other departments that handle areas such as facility management, payroll, accounting, and so on, also provide end user facing services. For this reason, other departments outside of IT may want to use Service Manager to handle these types of requests. To do so, they will need their own views, categories, support groups, and even requests published on the Self-Service Portal.

Many departments outside of IT are likely already handling incidents or requests by using some kind of tool to keep track of their work items. Smaller departments might use sticky

notes, Excel, Outlook, or a simple SharePoint list, whereas others might use third-party software. Service Manager can and should be used to handle many of these requests.

For example, departments such as Finance, Facility Management, and Accounting could realize several benefits of using Service Manager:

- End users will have a consistent experience every time they submit a request using the Service Manager portal. Organizations can also leverage email to capture requests in Service Manager so that end users are not even aware that their requests are being handled outside of IT.
- Using Service Manager as the only tool you use to handle requests streamlines the process, which can help reduce cost and improve user satisfaction.
- Categorization and reporting can provide weekly feedback to departments on the number of requests they have solved and the turnaround time as well. They can use this information to better manage and improve their handling of requests.

Implementing non IT usage of Service Manager is not more complicated than implementing it for IT purposes. It's just a matter of working with other departments, outside of IT, to determine the types of requests and services they are offering the business. The following scenarios show how some customers have implemented Service Manager for non-IT use.

Request a new credit card

One company implemented Service Manager to allow employees to request a company credit card. Before being moved to Service Manager, the process involved filling out an application form, reading the company-specific guidelines, and handing over the application to Finance for processing. Finance would then validate the information and send an email to the manager of the requestor for approval. After approval, Finance would order the credit card for the user. The requesting user often had no idea where in the process the request was, which generated many emails and calls to the Finance department for status inquiry.

Service Manager provided the following solution:

- The user accesses the Self-Service Portal and fills out a form for requesting a new credit card.
- The user's manager receives an email for approving or denying the request.
- The Finance department is notified after the request is approved by the manager and orders the credit card.

By offering a service request on the Self-Service Portal, the organization was able to immediately see the following benefits:

- True end user self-service is enabled because end users can return to the portal to check the status of their requests.
- The Finance department gains efficiency since a requester's manager contact

information is imported from Active Directory rather than having to be manually looked up.

- Managers can approve or decline requests via the portal or via mail, enabling them to handle approvals when they are not in the office.
- Additional workflow efficiencies are achieved since the Finance department was not involved in the request before the requesting user's manager had approved it. If the requesting user's manager denied the request, it is immediately closed, without involving the Finance department at all.

Request access to an invoicing system

This scenario involves creating offerings on the Service Manager Self-Service Portal to enable employees to request access to different line-of-business (LOB) applications, for example access to the invoicing system. Depending on the level of access of the employee and the type of request being made, manager approval may or may not be needed.

The following workflow in Service Manager was used to implement this scenario:

- The user accesses the self-service portal and fills out a form for requesting access to the LOB application.
- Depending on type of access, the user's manager receives an email for approval or denial.
- The team that manages the LOB application is notified of the service request via email.

The benefits of moving this workflow to Service Manager include:

- Users can create their requests and view the status of their requests on the portal.
- Automatically populating the approving manager saves time for the LOB application team.
- If the managers reject the request, the LOB application team does not have to be involved in the process.
- The manager can approve or decline the request via the portal or via mail, enabling the manager to handle requests even when not in the office.

Using Service Manager for customer service

The customer in this example is a reseller who services about 200 external companies.

The Customer Service Desk originally handled all issues and requests via email. The Service Desk had two or three people who managed all incoming emails, resulting in a cumbersome process for tracking the status of requests/incidents in their Outlook inboxes. The solution was to redirect the incoming emails to Service Manager, which then created incidents and service requests.

This Customer Service Department was the single point of contact for this organization. They in turn would forward emails for issues/requests to other departments for resolution. This is very similar to the function of an IT Service Desk, so it was an easy fit for Service Manager.

However, the deployment team did face some challenges with this implementation, such as identifying external customers without having them access the Self-Service Portal. In addition, the Customer Service Desk frequently sent documents and guides to customers, so the solution implemented had to handle this.

Since the external customers were not created as users in the organization's Active Directory, their contact and organizational information was not automatically imported through the Active Directory connector. The customer wanted the external customers in the Service Manager CMDB so the associated user would be linked to a request. To accomplish this, the Customer Service department periodically exported customer information from their CRM system and a custom connector was then created to periodically import this data into the CMDB.

Whenever the Customer Service Desk sent documents or guides to a customer, through a normal email, they also sent a copy to a monitored Service Manager-related mailbox. This ensured that any communication happening outside of Service Manager was added to the request.

This page left intentionally blank.

How to plan for Service Manager

Planning for Microsoft System Center 2012 Service Manager is critical since implementing Service Manager will have an impact on both the IT department that supports the organization and on the customers who subscribe to the services that IT provides. This chapter outlines the three most important areas to address prior to running setup. The most successful implementations of Service Manager begins with a vision of what the organization wants Service Manager to achieve. This vision must include the three areas that Service Manager will touch: the people who will use Service Manager to perform their roles, the underlying products that Service Manager relies on to populate the configuration management database, and the processes that Service Manager will automate.

It's not just the technology

You should begin your Service Manager journey by focusing first on your organization's people and processes to ensure that you consider the business needs when making the decisions about implementing Service Manager. Service Manager can provide considerable value by implementing an integrated set of IT processes and automation technologies based on Microsoft System Center, a highly integrated solution that helps IT organizations manage and drive efficiencies into very diverse and complex environments.

When implementing Service Manager, first establish a baseline of your current environment and situation. As illustrated by Figure 3-1, implementing Service Manager involves considering the 3Ps: people, processes, and products (tools and technologies). Understanding the current state of these 3Ps can help you see areas where improvements may need to be made.

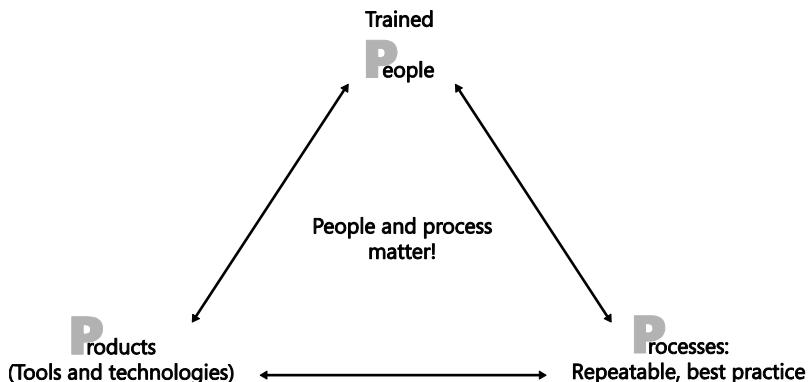


FIGURE 3-1 Understanding people, processes, and products are the key to implementing Service Manager effectively.

People

Implementing any tool in your IT department will have an impact on both the people who work in IT and the people who will use Service Manager. IT Service Management (ITSM) is all about behavioral changes: Attitudes drive Behaviors which result in Culture (ABC). Culture is only an outcome, so if a service and value culture is what your customer or organization is looking for, attitudes and behaviors need to change and adapt to the Service Manager journey.

Executive sponsorship combined with communication and user education is critical to the people aspect of your Service Manager implementation. How do you begin this journey? Like this:

- **Start with the "Why"** The first Why is why are we implementing a (new or replacement) service management solution. Second is, Why is Service Manager being deployed? What is the vision? Ensure all stakeholders not only understand the "Why" but also believe in it. If there is opposition, then the outcome, regardless of tools and processes, will be sub-optimal.
- **Understand the "What"** What processes will the Service Management implementation include? Will a phased approach be taken? What is in scope? What is out of scope? Gain commitment here and communicate and educate the "What" to all parties involved.
- **Implement the "How"** How will Service Management be used to manage ITSM processes and services? How will you configure and deploy Service Manager to support the "Why" and "What" elements?

Assigning customer roles and responsibilities is crucial to the effective implementation of Service Manager. Not getting all the right people involved and committed will put you at risk for experiencing poor adoption of Service Manager. If the key stakeholders and other critical people are not involved early on, they will feel that decisions were made in a vacuum, not

taking into consideration all of IT. Getting everyone on the same page is key, and everyone must participate. Also be sure to involve upper management to avoid snipers/spectators and ensure that the decisions made in workshops are defined and agreed upon by the process owners.

One common mistake is trying to implement all the Service Manager capabilities in one step. If you try to change more than 10 to 20 percent of operational behaviors in your organization with a Service Manager implementation, you risk poor adoption by the people who will be using it. Implementing Service Manager in small phases or by a staged process method is a better approach as incremental, ongoing, short-term improvements have been shown to have more success and sustainability than massive, organizational flips with a single-tool implementation.

Ensure all parties involved in the Service Manager implementation are empowered to make decisions. Employees must be empowered with responsibility, authority, and tools to succeed with their Service Manager journey. Executive sponsors must also empower the Service Manager project team.

Define critical success factors, key performance indicators, and metrics to measure staff, for example their compliance and contribution to ITSM programs. Define your success metrics early in the program and ensure that you measure them during the implementation and review and make adjustments as necessary.

View service management and improvement as a project with a definite end date. Service Management is a cultural change, so it takes time and can't be rushed. Continual service improvement is an ongoing exercise, and Service Manager provides the platform to enable this journey.

Another key element is training; never dismiss this as unimportant. Many organizations believe that their smart IT people will just know how to use Service Manager. They believe that their customers can go to the Self Service portal and figure out for themselves how to use Service Manager. This is simply not true, however, so defining a training plan that identifies all the audiences that will use Service Manager, creating the training by audience type, and then delivering that training to those who will use Service Manager are all essential. Carefully consider and plan both the initial training of your organization and the ongoing training for new hires. Such training must be part of the implementation and ongoing operations of Service Manager.

Process

The power of Service Manager is its ability to integrate and automate IT processes and technologies across both human and system resources in a consistent and coordinated fashion. To gain benefit from this, ITSM processes must be documented and understood by your organization. Unfortunately, many organizations haven't documented their processes adequately. Before implementing Service Manager, understand, document, and review the

current processes.

Figure 3-2 illustrates the Microsoft Infrastructure Optimization Model, which has been developed using industry best practices and Microsoft's own experiences with its enterprise customers. It is based on Gartner's Infrastructure Optimization Model (IOM), which in turn is based on MIT's Architecture Maturity Model. A key goal for Microsoft in creating the IOM was to develop a simple way to use a maturity framework that is flexible and can easily be used as the benchmark for technical capability and business value.



FIGURE 3-2 Microsoft Infrastructure Optimization Model for IT organizations.

The IOM helps customers understand and subsequently improve the current state of their IT infrastructure and what that means in terms of cost, security risk, and operational agility. Dramatic cost savings can be realized by moving from an unmanaged environment toward a dynamic environment. The goal of Service Manager is to have the IT Infrastructure move from a highly manual and reactive state to a highly automated and proactive state. Also the processes will move from fragmented or non-existent to optimized and repeatable through documentation and automation with Service Manager. The end result is that the IT organization will improve their business agility and deliver business value increases as they move from the Basic state up toward a Dynamic state, empowering information workers and managers and supporting new business opportunities.

Product

Most Service Manager implementations involve the replacement of an existing ITSM tool. Many organizations already own Service Manager as part of their System Center licensing agreement, and they can therefore see moving to Service Manager as a potential cost savings measure. Unfortunately, many organizations also try to make Service Manager act and behave like their current ITSM tool without examining the business needs and service management requirements of their organization.

The following list outlines some of the product issues and challenges that organizations face when implementing Service Manager:

- **Deploying Service Manager without specific objectives** IT systems, including ITSM systems, should meet a clear set of objectives. The first question to be answered is, Why are we implementing Service Manager? This allows for the objective determination of success or failure and allows for decisions to be made before large sums of money are spent on deployment.
- **Looking for Service Manager to meet a functional list, for reasons of cosmetics or neatness, or to ensure ITIL compliance** Technology is intended to underpin and automate some or all of an organization's processes. There should be both test of efficiency and test of effectiveness measures that support the business mission.
- **Implementing Service Manager without assigning post-project resources and funds** Service Manager requires maintenance and ongoing support and improvements to meet the organizational needs for the ITSM journey.
- **Focusing on configuration management without a service context** Perhaps one of the biggest areas of failure in ITSM is configuration management. The value runway is very long, it is typically discovery-only focused, and often it does not align to services and capabilities of the organization. A successful CMS/CMDB answers the questions of accountability (who, what, when) and relationships. Start with the out of box capabilities of the Service Manager CMDB. By linking configuration items (CIs) to work items in Service Manager, you will quickly improve your incident and change management processes and enable problem management.

Customers who implement Service Manager to replace an existing service management tool often focus on phone-based ticket creation and are thus typically disappointed with Service Manager. These old service management tools view everything as a ticket and really don't focus on enabling end users to perform self-service based on a service model. Service Manager enables service-based and automated intake where you need to focus on ruthless standardization, extreme automation, and separation of service requests from incidents. Service Manager therefore does not provide some of the traditional call center capabilities like decision call tree, launchpad for initial intake (before you know what service, incident, request), or quick ticket (although this can be accomplished through templates).

Knowing this, you need to take caution when implementing Service Manager so as not to recreate side-by-side functionality with your existing service management tool or try to make Service Manager act and behave like the tool that you are trying to displace. The discussion should start with why you need Service Manager as a tool. Why are you replacing the incumbent tool? Is it too costly to maintain due to customizations? Is reporting difficult or nonexistent? Are you doing it for cost savings? Or are you just not happy with the current tool?

Next, you must examine "What" outcomes that management needs to achieve with Service Manager. You need to address the business requirements first. Focus on outcomes or "What" the process needs to achieve first; step away from the *methods* discussion on how Service Manager actually does this. The "What" discussion should focus on what Service Manager needs to achieve to make the IT organization successful. Drive the discussion by understanding, documenting, and agreeing first on what are the desired process outcomes. Once these "What" requirements have been documented and agreed upon, move forward with the "How," planning how Service Manager can achieve these requirements. Service Manager has been most successful for customers when they:

- Don't try to reinvent past toolset behaviors and capabilities in Service Manager. Instead they leverage how Service Manager works, specifically the benefits of using the Self Service portal.
- Don't try to implement existing tool capabilities feature by feature in Service Manager. Implementing Service Manager provides a great opportunity to examine the business needs, the processes, and underlying technology to provide better value to the business and the end users.
- Move away from using the term tickets and instead enable service-based intakes via the portal. This enables better routing of requests and incidents since you are leveraging services for ownership and resolution.
- Leverage business services in Service Manager and the importing of distributed applications from Systems Center Operations Manager.
- Use a service catalog on the portal to make navigation and organization of services provided to the business available for self service and end user enablement via automation.

Implementation roles

Table 3-1 lists the roles needed for engagement of the appropriate people in a Service Manager deployment. Not having the appropriate people engaged and involved in the decisions that are made to implement Service Manager puts you at risk for alienating supporters in your organization, when they feel they were not consulted or informed about what decisions were made during implementation. Marketing and communication are key

since everyone in IT will be affected by the implementation, and they must feel that their input was taken into consideration.

The list covers all of the capabilities in Service Manager. Make sure there is a name assigned to each role plus an outline of the time required for these roles. Remove the role types for the capabilities you will not be deploying.

TABLE 3-1 Roles that need to be engaged in a Service Manager implementation

ROLES/ROLE TYPES	RESPONSIBILITIES
Project roles	
Executive Sponsor	Executive who is the key stakeholder of the Service Manager deployment.
Project Lead/Manager	Customer-appointed project manager.
Training Lead	Responsible for the creation of training material for Service Manager. Even IT staff need to be trained on proper use of Service Manager for deployment and ongoing training needs for IT.
Communication Lead	Creates the communication plan and ensures timely communication on project status and next steps.
Reporting Lead	Will own the reporting requirements and ensure that reporting is managed after the engagement ends.
Process roles and subject matter experts (SMEs)	
Incident Management	Owns the incident management process and makes the Service Manager tooling decisions pertaining to incident management; include the help desk and tier 2 and tier 3 resources.
Problem Management	Owns the problem management process, makes the Service Manager tooling decisions on how problem management is performed.
Change Management	Drives the tooling requirements in Service Manager; they understand the approval processes and what types of changes need to be implemented in Service Manager.
Release Management	Release managers can also include project managers as they really know how release management works in the organization.
Service Request Fulfillment	Service requests could extend outside of IT, to include facilities, HR, and other groups needed to fulfill service requests and to understand how service requests are completed.
Service Level Management	Discussions around what SLAs are in place will require service level managers, and, if they own the service catalog, they will drive the service catalog on the Service Manager portal.
Configuration Management	The configuration management process owner will drive what decisions regarding the management of configuration items in Service Manager.

Technology roles	
SQL SME	If you are going to need assistance with SQL for the deployment of Service Manager, be sure to engage an expert early in the project lifecycle.
Exchange SME	Exchange connector and SMTP needed for Service Manager.
AD SME	For assistance in understanding how Active Directory is deployed and how to configure the AD connector for optimal data import.
SCOM SME	To understand how Operations Manager is deployed, what management packs and distributed applications are in the environment to import.
SCCM SME	For expertise on how Service Manager can be deployed and whether asset intelligence has been deployed correctly to get the correct CI information.
SharePoint SME	You need to deploy the Service Manager portal, and if you need customization of the portal, this person is your best resource.
SCSM Infrastructure Lead	This person is going to be attached to your hip throughout the deployment of Service Manager. They need to do the bulk of the work during deployment as they will be responsible for managing Service Manager when you leave. Ensure that this person is available to you for the entire time you are on site.

How to prepare for a Service Manager installation

When implementing Microsoft System Center 2012 Service Manager, you need to ensure that the quality of data to be imported and the maturity of Service Management processes are well understood and defined. Service Manager will provide value only if you understand the IT processes, organization and infrastructure prior to implementation.

Many organizations are disappointed in what Service Manager has to offer when they have not taken the necessary steps to document their processes, understand their people requirements, and clean up their data sources before installation and configuration of Service Manager. One of our favorite sayings here at Microsoft is "Service Manager is not a laundromat," meaning that the quality of the data, processes, and people requirements going in should be in an optimal state prior to running setup. This chapter explores the key technology and process considerations you must explore before deploying Service Manager in your environment.

Technology considerations

It takes more than just reading the prerequisites documentation to determine if you are ready to implement Service Manager in your environment. The quality of the information in the data sources you will use to populate Service Manager is critical to ensure that the Service Manager Configuration Management Database (CMDB) contains the necessary data elements to support the service management goals of your organization. Service Manager relies on data sources to populate the CMDB, and if these sources are not well managed, importing data into Service Manager will only mirror the poor state of your Active Directory, System Center Configuration Manager, and System Center Operations Manager deployments instead of adding the value needed to support your service management vision.

Active Directory

The Active Directory connector imports users, groups, computers, and printers to populate the CMDB. If your Active Directory objects are not well managed, for example if you have users in Active Directory who no longer work for your organization, importing this data into Service Manager will bloat the CMDB with incorrect information. Prior to importing data from Active

Directory, you should review the considerations listed in Table 4-1 and ensure that you examine the quality and management of data coming from Active Directory.

TABLE 4-1 Service Manager considerations involving Active Directory

ACTIVE DIRECTORY DECISIONS	WHY THIS IS IMPORTANT FOR SERVICE MANAGER
Are the user fields populated for each Active Directory object?	Service Manager imports many fields from Active Directory. Like email, department, location, phone numbers, and managers are all used by Service Manager for contacting users, for example, the location information can be used to build out assignment workflows for support groups, and the Manager field is useful for approvals of certain types of requests. In addition, large organizations with multiple locations must know where an affected user is located for deskside support.
Are all computers and printers registered in Active Directory?	Having computers and printers registered in Active Directory enables the linking of incidents and service requests to the actual configuration item, which enables better reporting and better identification of known errors in the environment.
Are old users, computers, and printers moved to a different OU?	It is recommended to move old users, computers, and printers to a different OU that can be excluded from the Active Directory connector to avoid importing these items into the production Service Manager database.
Is there a standard naming convention for Active Directory objects?	Having a consistent naming convention for computers and printers makes it easier to identify these items in Service Manager and link these items to work items.

Operations Manager

There are two connectors for Operations Manager to import configuration items (CIs) into Service Manager. In addition, the Operations Manager Alert connector can import active alerts and create Incidents in Service Manager. If Operations Manager is not managed, turning on this connector for all managed objects will cause incidents to flood into Service Manager. Prior to implementing the Operations Manager connectors, examine the health of Operations Manager and understand what management packs are deployed and which items are monitored. Table 4-2 lists some Operations Manager areas to be reviewed.

TABLE 4-2 Service Manager considerations involving Operations Manager

OPERATIONS MANAGER CONNECTOR CONSIDERATIONS	WHY THIS IS IMPORTANT FOR SERVICE MANAGER
What is your alert-to-ticket ratio?	The alert-to-ticket ratio defines the number of alerts that actually constitute an incident. Poorly tuned Operations Manager environments will have many alerts that are ignored because the thresholds have not been properly defined in Operations Manager. People become desensitized to a flood of alerts coming from Operations Manager, and if they are imported into Service Manager, you will find multiple incidents in the Service Manager database with no ability to act upon these open incidents.
How many alerts do you get per day?	A high number of alerts usually means that no tuning has been performed on Operations Manager.
Is there an owner of the Operations Manager environment?	If there is no ownership or management of Operations Manager, do not use the Alert connector. Since management rarely looks at the alerts in Operations Manager, turning on the connector to create incidents per alert will be overwhelming if Operations Manager is not tuned or managed.
How many servers are monitored by Operations Manager?	If not all critical systems are monitored by Operations Manager, you won't see all the critical systems that IT wants to manage in Service Manager.
What management packs are deployed in Operations Manager?	Knowing what management packs are deployed will enable you to understand what systems are managed and what services can be managed in Service Manager.
Can you map the management packs to the IT support groups that need to respond to the alerts?	Start by listing all the Operations Manager management packs and identify the IT support groups that need to action these alerts. You can prebuild templates based on the management packs and define the support groups that need to action the incoming incidents in Service Manager, thus reducing the bloat of open incidents with no defined support group.
Have you created any distributed applications in Operations Manager?	Distributed applications in Operations Manager are imported as business services in Service Manager. This can add value by linking incidents, changes, and service requests to business services since this adds a deeper understanding of the services that IT supports.

Configuration Manager

Service Manager ships with two connectors for Configuration Manager. One of these is the Configuration Item connector, which pulls in hardware inventory, software inventory, and software updates. To get the most out of this connector, you will need to enable asset intelligence in Configuration Manager and ensure that the processor data, software inventory, and licensing WMI classes are all enabled. The data imported from Configuration Manager will augment the existing data pulled from the Active Directory connector and link the user with the computer by leveraging the asset intelligence top console user as the primary user of the computer.

The second Configuration Manager connector is the Desired Configuration Management connector, which works with the Desired Configuration Management (DCM) feature of Configuration Manager to raise incidents in Service Manager when the managed configuration drifts. Before you implement these Configuration Manager connectors, you should review the health of the environment as described in Table 4-3.

TABLE 4-3 Service Manager considerations involving Configuration Manager

CONFIGURATION MANAGER CONNECTOR CONSIDERATIONS	WHY THIS IS IMPORTANT FOR SERVICE MANAGER
Is there a primary person responsible for Configuration Manager?	If no one person is responsible for Configuration Manager, chances are it is not managed.
How many primary sites exist?	Creating one connector instance for each primary site improves the speed and efficiency of imports.
Are there any empty collections?	Do not point a connector at empty collections and be sure to import only relevant ones.
How many objects are managed by Configuration Manager?	The initial sync for Configuration Manager will take time. For example, ten thousand computers will take several hours, so don't do the initial sync during the day.
Is there a process for deleting old computers from Configuration Manager?	Importing old computers just bloats the Service Manager CMDB. Create collections that have filtered out old computers before you perform an import. Configuration Manager 2012 allows filtering of computer objects.

Virtual Machine Manager

The Virtual Machine Manager connector imports library data, such as virtual machine templates, storage classifications, and service templates, that can be used by service requests in Service Manager. If Virtual Machine Manager also pushes discovery data into Operations Manager, you also need to create an Operations Manager CI connector to import these data

into Service Manager. You must make sure that Operations Manager integrates with Virtual Machine Manager first and that the Virtual Machine Manager management pack is imported into Service Manager as well.

Process considerations

Besides the technology considerations described above, preparing for a Service Manager deployment also involves thinking about processes such as incident management, problem management, change management, release management, and service requests.

Incident management

Most organizations who deploy Service Manager implement incident management functionality in the product. One key benefit of this is the ability to integrate Operations Manager alerts as incidents in Service Manager. For best results, you will need to include your service desk, the incident management process owner, and tier 2 and tier 3 support in the discussion and decision making around incident management. Some of the key areas to examine and discuss before configuring incident management are listed in Table 4-4.

TABLE 4-4 Services Manager considerations involving incident management

INCIDENT MANAGEMENT CONSIDERATIONS	WHY THIS IS IMPORTANT FOR SERVICE MANAGER
Is the incident management process documented, with roles, policies, and activities?	A common documented incident management process with defined roles and activities will simplify and provide a consistent experience within Service Manager.
Are there defined roles, responsibilities, and activities for incident management?	This information is necessary to perform the physical tooling of Service Manager incident management.
Is Service Monitoring currently integrated into your ITSM tool?	Integration of Operations Manager into Service Manager incident management will enable IT to proactively identify issues before they become critical. The historical alert data can also be tracked as incidents and be used for reporting, trend analysis, and problem management.
Are CI data and services linked to incidents?	Linking CIs and services to incident records provides additional context of the type of incidents affecting the organization and adds an extra dimension to incident management reporting.
Do you distinguish between incidents and service requests?	Incidents represent failure demand and service requests represent value demand and should be managed separately.

What information needs to be logged in an incident?	Verify that these requirements are mapped to the fields in Service Manager. Will you need to modify labels on fields? Add more custom fields? Try to limit the number of form customizations and ensure that the need for a new field is well planned.
Is there a major incident process?	Ensure that this is understood and that you walk through a case scenario for a major incident so you understand what workflows, notifications, and teams need to be involved.
What incident management reports need to be produced?	Understanding what needs to come out of Service Manager in reports will drive what needs to be entered and tracked in Service Manager to produce the necessary reports.

Problem management

Problem management relies on incident management data to identify trends in recurring incidents as well as patterns of service issues. You should implement problem management after implementing incident management. The Operations Manager Alert connector provides additional incident information, which is valuable input for the problem management process.

You can manage problems in two ways. One way is to do it reactively, usually triggered by a major incident or service outage. The other approach is proactive problem management. This is when the problem analysts review recurring closed incidents to find patterns that can help determine the root cause of recurring incidents. Most organizations start with reactive problem management. Then, as more resources and time are spent on problem management, they begin to move to a more proactive approach. Table 4-5 outlines the discussion areas and decisions that need to be reviewed in Service Manager.

TABLE 4-4 Service Manager considerations involving problem management

PROBLEM MANAGEMENT CONSIDERATIONS	WHY THIS IS IMPORTANT FOR SERVICE MANAGER
Is the problem management process documented, with roles, policies, and activities?	A common documented problem management process with defined roles and activities will simplify and provide a consistent experience within Service Manager.
Is incident management implemented within Service Manager?	Problem management needs incident data for trend analysis and root cause detection. Using Service Manager for incident management is required for problem management.
Is Operations Manager alert connector currently enabled in Service Manager?	Integration of Operations Manager into Service Manager incident management will enable IT to proactively identify issues before they become critical.

	The CIs with the most incident reports in problem management reporting are a good indication of which managed objects are generating the most alerts. This data can be used to perform problem management.
What is the quality of the incident data in Service Manager? Is incident data reviewed for quality and correctness?	Poor incident data means worse problem management data. Regular reviews of incident data to ensure that configuration items are linked to incidents, that proper incident classifications are used, and that proper symptoms and error messages are tracked in Service Manager results in better data to review for the creation of problem records.
Is there a root cause analysis methodology?	A documented, known approach to root cause analysis will help reveal root causes of problems. Not having a repeatable approach used by all problem analysts causes makes it harder to determine the root cause of a problem.
Do you plan to leverage knowledge articles for workarounds and solutions to problems?	Leveraging knowledge articles with workarounds and solutions for common incidents and problems enables IT support and end users to resolve their issues quickly.

Change management

Change management in Service Manager works best if you first create predefined change templates that analysts can select from a list. A change request consists of multiple activities built into a single change. Service Manager change management includes out of box change templates that demonstrate the different change categories available. Standard changes are pre-approved changes and do not require approval, but they still need to be logged in order to track that the change was performed. Table 4-5 lists some considerations in this area.

TABLE 4-5 Service Manager considerations involving change management

CHANGE MANAGEMENT CONSIDERATIONS	WHY THIS IS IMPORTANT FOR SERVICE MANAGER
Are the change management process, procedures, and activities documented and shared?	A documented and defined change process with defined roles and activities simplifies the process and provides a consistent experience within Service Manager.
Are changes currently linked to CIs and services?	Having changes and their related activities linked to CIs and services enables better impact and risk analysis and provides information to the other processes in Service Manager. For example, when an incident occurs in the Messaging Service, looking up the Messaging Service will show that there are changes open against that service.

Who in the organization can log changes?	Most changes are logged by IT and usually by the person who wants to perform the change. Discuss who will be logging changes in Service Manager and understand the scope of change management.
Do you have defined standard changes?	Standard changes do not require approval and are ideal for creating change templates in Service Manager.
Do you have predefined change approvers? How is the approver determined?	Change approvers can be defined by service, configuration item, or by organizational structure. Knowing who needs to approve changes and how they will interact with changes in Service Manager needs to be determined.
Is there a change advisory board (CAB)?	Who are the members of the CAB? Are they a static group? What type of changes does the CAB review? How are changes approved...unanimously? These items are key to determining how to manage approvals in Service Manager and how to determine voting logic.
Is there post implementation review (PIR) on changes? When and why?	If the organization performs PIR on changes, this needs to be set as an activity on the change template. Are PIRs only performed on failed changes? Then a notification will need to be defined for failed changes. Changes that are successful should also be reviewed to find best practices for change implementations.
What change management metrics does the organization review?	Knowing what needs to be in a report is useful for understanding what needs to be logged into a change record. Review the out of box reports for change management to ensure that these reports are sufficient for change process performance reporting.

Release management

Service Manager supports the release to production process. Release management in Service Manager supports the grouping of approved changes to be tested and deployed together in a single release to take advantage of maintenance windows and reduce the number of times a particular CI or service has single changes applied. To take advantage of release management in Service Manager, you first need to have change management deployed and in production. Release management uses activities and changes to plan the effective use of deployment resources to implement multiple changes into one release package.

Including project managers (PMs) in the requirements definition is crucial for success. Many PMs understand the bigger picture of project releases, and they will provide valuable input into how the process of release management actually works. Table 4-6 outlines the questions

that need to be asked to determine the appropriate requirements for release management in Service Manager.

TABLE 4-6 Service Manager considerations involving release management

RELEASE MANAGEMENT CONSIDERATIONS	WHY THIS IS IMPORTANT FOR SERVICE MANAGER
Is the release management process, procedures, and activities documented and known?	A documented and defined release management process with defined roles and activities simplifies the process and provides a consistent experience within Service Manager. Also, knowing what is considered a release in your organization is worth noting prior to implementing release management in Service Manager.
What type of releases are performed?	In Service Manager, release type is defined as planned or unplanned, but knowing if there are different types of releases in your organization is critical for reporting purposes.
What categories of releases does your organization perform?	Categories of releases add granularity for reporting and tracking. Examples of release categories are bug fixes, projects, and new services.
Who in your organization is responsible for building release packages?	These people will have a good view on how release management currently works and also will need to be engaged during the build and testing phases of release management.
How are changes currently grouped into a release?	It is important to know how changes are grouped and if they are based on configuration items or services.
What release management metrics does the organization review?	Knowing what needs to be in a report is useful for understanding what needs to be logged into a release. Review the out of box reports for release management to ensure that these reports are sufficient for process performance reporting.

Service request management

Service requests are requests for existing, preauthorized services and features that Service Manager can manage as a work item. Service requests are tightly coupled with the service catalog to facilitate service request management. Table 4-7 describes some considerations involving service requests.

TABLE 4-7 Service Manager considerations involving service management

SERVICE REQUEST CONSIDERATIONS	WHY THIS IS IMPORTANT FOR SERVICE MANAGER
Are the service request process, procedures, and activities documented and shared?	A documented and defined service request process with defined roles and activities simplifies the process and provides a consistent experience within Service Manager
Who performs service requests in your organization?	If other groups in the organization (outside of IT) complete service requests, they will need to use Service Manager. Including non-IT requests extends the value of service requests on the portal.
What services are offered to end users	The Service Manager portal organizes items into service offerings. These need to align to the service catalog and be published in languages that the business and end users understand.
Are there different SLAs for different types of service requests?	You need to understand these SLAs to trigger workflow based on what type of service request is chosen.
What are the sources of service requests?	Do end users call the service desk, fill out web forms, or email their requests? Understanding the sources of service requests provides an input mechanism/choice for how Service Manager will handle the requests.
Are service request statuses and implementation results tracked? If so, what results need to be tracked?	Service request implementation results are used for reporting and management reviews, so you will need to understand what is currently tracked for service requests and ensure that this information can be captured and reported.
What support groups are involved in the completion of service requests?	Do you need to include groups outside of your immediate team/IT to complete service requests? How are they integrated into the process and what are the plans to migrate and move them to Service Manager? If there are groups outside of IT, remember to include them in parts of request design and training in how to use Service Manager.
Can customers change the urgency of service requests?	Can or should Service Level Targets be changed for already submitted service requests, just because a customer finds it more important than normal? If a customer can modify the urgency of service Requests, that might affect the KPI for the servicing team. You need to be aware of what you track and report on for Service Level Targets..
Who owns the service request process?	The process owner owns the tooling decisions in Service Manager since they are accountable for the outcome of the process.

Management packs

One of the first things customers realize when implementing Microsoft System Center 2012 Service Manager (if they didn't know already from using Operations Manager) is the need to structure, name, and manage Service Manager management packs.

Most Service Manager configuration changes are contained, transferred, configured, authored, and customized through the use of management packs, so properly structuring and naming them are of the utmost importance before beginning to create new ones.

Management pack general guidance

The following guidelines can help you develop a standard and structured method for naming and combining modifications to create custom management packs. Note that these are only guidelines and not rules you must follow:

- Where possible, do not store anything in the Service Manager out-of-the-box (OOTB) unsealed management packs. Best practice is always to change these to custom management packs using the naming conventions detailed in the next section.
- When extending classes, keep extensions separate from form modifications so the form can be removed (deleted) from Service Manager without impacting the class or data records. Class extension management packs must be sealed as a prerequisite for transferring the data in them into the Data Warehouse.
- Form management packs must be sealed when they reference a class extension. Furthermore, the Form management pack also must be sealed for templates to be created from it.
- Be sure to mark or indicate management packs that include manual edits, either by name or description, so that it's clear that the management pack was manually edited.

Management pack naming guidance

The following commonly used naming convention follows Service Manager management pack naming conventions. These examples are from Service Manager 2012.

- **ServiceManager.IncidentManagement.Configuration.XML** Unsealed Management pack for incident related configuration settings (OOTB lists).
- **ServiceManager.IncidentManagement.Library.MPB** Management pack bundle for all sealed incident management packs.

While not required, following a naming convention consistently can certainly be helpful in the administration and configuration of Service Manager. It is therefore recommended that a similar naming convention be adopted within your organization when you make changes to management packs and when you create management packs during configuration.

When you develop your own naming convention, a good naming convention for Service Manager might be **CC.MMMMMMM.FFFF.NNN.NC** where:

- **CC** Company/organization (short two or three characters), for example MS for Microsoft.
- **MMMMMM** Module (process), management function, service, or tech area. Some possible standard abbreviations are IR, CR, SR, PR, HumanRs, CustSvc, Network, Desktop, Server, and so on. You should standardize this list across your implementation and organization.
- **FFFFF** Function or purpose of modifications, for example email, notification, workflow, views, config, tasks, service, and so on. You should standardize this list across your implementation and organization as well.
- **NNN** Additional coding specific to the modification (optional).
- **NC** Add "NC" if the management pack has been modified via XML edit and may create issues if modified via the console. This way it can be immediately noted that the management pack is not to be edited or modified from the console. The suffix NC should be added to only the management pack display name and not to its internal name. It is more difficult (and sometimes impossible) to change the internal name of the management pack, and it will always be visible in the console while the internal name may not be.

Bundling modifications

Choosing between putting everything in a single management pack or using numerous management packs with individual modifications in each will not keep you from making bad decisions. Neither a single management pack nor thousands of management packs is the correct answer; the best approach is somewhere in the middle. Consider the following when deciding on the best approach:

- Who will be authoring the management packs? Will different groups have their own set of management packs to manage?
- How will modifications be promoted into the production environment? By release package? Larger multi-function modifications have greater risk in a release implementation.

- Some management pack modifications may require change management; others may not (adding list value).

Some types of changes that might be bundled into a single management pack include:

- Process roles
- Process role views by process (may require type projections)
- Process role tasks by process
- Technical, service, or process groups and queues
- Service level objectives, metrics, and calendars
- Module enumeration list items
- Service/support group form templates (dictate activities to be assigned)
- Service offering, its request offerings, service request and incident templates
- Workflow plus notification (must be together because they are stored together)
- Views that use a folder or a type projection that isn't in a sealed management pack (must be located in the same management pack)

Some examples, then, of an incident class extension might include:

- MS.Incident.ClassExtension (sealed)
- MS.Incident.ClassExtension.Form (sealed, for use with new class extension)
- MS.Incident.ClassExtension.Configuration (unsealed, to be used for capturing configuration settings such as enumerated lists of the class extension)

Some items that should be in their own sealed management pack (not bundled) for use across the system might include:

- Type projections, so they can be referenced freely by any view in any management pack
- Custom images, to be referenced remotely by the views
- Folders, if you want to split up the views inside that folder between different management packs

Naming and bundling views and view folders

Views provide a way to limit, sort, group, and display work items and configuration items within Service Manager. Aside from forms, views is where most system users select, observe, and take action on information within Service Manager. Therefore, management pack names and display names may be different. While management pack names for views should follow naming standards as indicated earlier, views should have user-friendly names that make sense and are understandable to the audience acting within the processes.

Folders and type projections should be placed in sealed management packs so that type projections can be referenced by views and e-mail templates and don't need to be duplicated

elsewhere. Folders should also be in sealed management packs so they can store views from different management packs in that folder.

Naming and bundling templates

The number of templates increases as procedures mature in the Service Manager environment. As a result, ruthless standardization and normalization is recommended to reuse and modularize templates across as many groups as possible. However, there will be differences among teams because templates may also be edited by respective teams, rather than by Service Manager administrators, so the team's management packs will need to be imported separately when updated. For example, a management pack for all service requests and form templates to the network team might be MS.ServReq.Template.NetworkSupport.

Naming and bundling service offerings and request offerings

Service offerings and request offerings (which mimic a one-to-many relationship) should be contained in management packs aligned with a service offering management pack. For example:

- **MS.Services.NetworkSupport** Management pack for all internal service offerings and request offerings of the network team
- **MS.Services.DataCenter** Management pack for all internal service offerings and request offerings associated with data center services

The request offering and associated template must be in the same management pack. Although not required, containing the request offering and the service offering in the same management pack is best practice.

Naming and bundling groups and queues

Groups and queues are variously based on process, service, technology, and priority/service level. Some examples include:

- **MS.Incident.Queues** All incident queues defined
- **MS.Incident.Groups** All incident groups defined
- **MS.Configuration.Groups** All configuration item groups (e.g., would include where "Global Operators" would be stored)

Naming and bundling console tasks

Tasks are held in management packs separately from user roles. Store tasks in a sealed management pack by themselves, and name them based on the object they affect. For instance:

- **MS.SR.ConvertToIR** Console task to convert a service request to an incident.

- **MS.System.ExportAllIMPs** Console task to export all management packs.

Naming and bundling notification templates and subscriptions

Since notification subscriptions are dependent upon notification templates, they should be maintained together by module. For example:

- **MS.Incident.Notifications** Notification subscriptions and templates for incident management
- **MS.ServiceRequest.Notifications** Notification subscriptions and templates for service request management

IMPORTANT Notifications can be configured from both subscriptions as well as the workflow\configuration view. In Service Manager 2012, subscriptions handle related recipients as well as periodic notifications. The only time it is recommended to configure notification type workflow within the workflows\configuration view is if it is a notification specific to a particular workflow that also applies a work item template that doesn't have any overlap with any other processes or workflows.

Sealing management packs

The following items should be in their own sealed management pack for the reasons indicated:

- **Custom classes** These should be sealed so you can:
 - Report on new classes
 - Build forms for new classes
 - Create views, templates, notifications, and other configurations for new classes
- **Custom forms** These should be sealed so you can create templates from them. Also note that although you don't have to build a custom form for a custom class since Service Manager will provide a generic form that contains all the properties, most of the time you'll want or need to create your own.
- **Class extensions** These should be sealed so the new properties/relationships can be reported and so you can add them to a customized form.
- **Existing form customizations** These should be sealed so you can create templates for them.
- **Type projections** These can be referenced by views, e-mail templates, and other items. It is often best to let these be accessed by any management pack in the system so you don't need to duplicate them.

- **Management pack bundles** When you bundle a management pack, you are adding a .dll, image, or other reference file along with the import. If you do not seal the management pack, you won't be able to reference these resources from outside the management pack. This is most commonly an issue with images. If you bundle an image with an unsealed management pack, only that unsealed management pack can reference that image for use. Also, if you need to export that unsealed management pack to make changes to the XML, you cannot reimport it without re-bundling it with the image first. Therefore, it is best to import just a blank, sealed management pack bundled with the image, then reference that image in your unsealed view management pack.
- **Folders** Sometimes it makes sense to store a folder in a sealed management pack because it will let you store views from different management packs in that folder.
- **Data warehouse extensions** These need to be sealed to extend the data warehouse. If they are unsealed, the MPSyncJob won't synchronize them with the data warehouse databases and your extensions will not be created.

Sealing management packs requires a strong name assemblies (SNK) file that contains a unique key pair consisting of a private and public key. For more information on SNK files, see <http://msdn.microsoft.com/en-us/library/wd40t7ad.aspx>.

To seal a management pack you can use either the Authoring tool that comes with Service Manager or the FastSeal.exe tool. For more information on using the Authoring tool, see the step-by-step guidance at <http://technet.microsoft.com/da-dk/library/hh495605.aspx>. For additional information on sealing management packs and to download the Fastseal.exe tool, see <http://blogs.technet.com/b/servicemanager/archive/2009/12/25/sealing-management-packs.aspx>.

To use FastSeal you might first need to make some changes to the XML. If the management pack has a category, you will need to make sure you add a line for the public key token. Here is an example of a category for a management pack with a public key token:

```
<Category ID="Category.xyz" Value="EnterpriseManagement!XYZ">
<ManagementPackName>MS.Change.FormCustomization</ManagementPackName>
<ManagementPackVersion>7.0.6555.0</ManagementPackVersion>
<PublicKeyToken>31bf3856ad364e35</PublicKeyToken>
</Category>
```

Sometimes you will have to use FastSeal, for example if your management pack breaks any of the Authoring tool's rules.

Updating a sealed management pack

If you have an unsealed version of a management pack, you can make changes to the management pack and then seal it with the same SNK file used to seal the original

management pack. Make sure the version number of the new sealed management pack is greater than or equal to the existing sealed/imported management pack; the authoring tool will do this automatically when you seal it. Once this is done, you will be able to import the new sealed management pack and replace the existing sealed management pack. Note that Service Manager will still warn about potential data loss on import, due to the older version being installed.

If you don't have the current unsealed management pack, but you have the SNK file that you used to seal the original management pack, you can export the sealed management pack from the console to generate an unsealed XML version of the management pack. You can then use the unsealed management pack to make changes, reseal it, and import it as explained above.

There are some cases when neither of these methods will work, for example with form customizations. With these types of management packs you must delete the existing management pack before importing the updated management pack. Note that anything stored in a management pack will be deleted when the management pack is deleted. This is especially important with regard to class extensions because any data stored in the class extensions will also be deleted and can't be recovered even if you reimport the management pack. That is why you must always keep forms and class extensions in separate management packs.

Versioning management packs

Management pack versioning can be challenging because it has to be done manually, and most users of Service Manager seem uninterested in doing this. The key to controlling the version numbers of management packs is defining what each segment of the version number means to your organization and then staying consistent with this. You can use industry standards for this purpose, but management packs are not source code, so coming up with your own definitions that are specific to Service Manager is usually a better option. A typical versioning model might look like this:

1. **Major** Follows version number of the Service Manager environment (7 for 2010 and 2012)
2. **Minor** Major changes (not really relevant since this number won't change much for most clients)
3. **Build** Changes (add/remove functionality)
4. **Revision** Bug fixes

You can change the version of the management pack in the authoring console or by directly editing the XML. You will find <Version> in the management pack in the section named Identity as follows:

```
<Identity>
  <ID>ServiceManager.IncidentAssignmentChanges.Notification</ID>
  <Version>7.0.5244.0</Version>
</Identity>
```

Backing up management packs

It's always a good idea to keep a backup of all your work. As far as backup of management packs is concerned, the sample script below demonstrates how to back up all unsealed management packs in your environment. Note that this should not be used for disaster recovery purposes since this needs to be performed at the database level and therefore should not require reloading of management packs.

```
import-module smlets

# format directory name into yyyy-mm-dd format
$directory = get-date -uformat %Y-%m-%d

# Create Directory in MP backup path
New-Item -type directory -path "C:\Custom\MPBackup\$directory"

# Store unsealed management pack into variable
$MP = Get-SCManagementPack | where {$_.Sealed -eq $False}

# Export unsealed management packs to the directory
$MP | Export-SCManagementPack -TargetDirectory "C:\Custom\MPBackup\$directory"
```

The above script can be configured to run on a daily basis as a task in the console and simply places all unsealed management packs into a subdirectory named with the current date.

Renaming management pack filenames

To have meaningful filenames, you might need to rename the filename itself and also the three instances where it appears in the XML. Be sure to delete the old management pack from your dev/test system before renaming it.

IMPORTANT Service Manager determines whether a management pack already exists by examining its internal name. So if you change its internal name, it will be treated like a brand new management pack. This means if you import a management pack after changing its name, at best you'll get an error message and at worst you'll create duplicate configurations. That is why it is important to delete the existing management pack before importing the renamed version. This means it is sometimes impossible to rename management packs after a system has been in production, since deleting certain management packs can cause loss of production data. So it's important to get the naming correct before any custom management pack is imported into the production system.

In addition to the filename itself, the following sections of the management pack will need to be in sync with the filename (which cannot have spaces, special characters, and so on):

- Manifest ID
- Category

The following example shows where you'll need to change the management pack ID and Name multiple times when you rename a management pack:

```
<Manifest>
<Identity>
<ID>ManagementPack.76e90f21466e4b35b1e1544df625fa78</ID>
<Version>7.5.1561.1</Version>
</Identity>
<Name>SendMail.Configuration</Name>
...
...
<Categories>
<Category ID="Category.26203af0ebac40b4a64f75905b4f0bee....>
<ManagementPackName>ManagementPack.76e90f21466e4b35b1e1544df625fa78...</ManagementPackName>
<ManagementPackVersion>7.5.1561.0</ManagementPackVersion>
</Category>
```

This page left intentionally blank

Optimizing the Service Manager environment

Microsoft System Center 2012 Service Manager is a complex product that requires Windows Server, Internet Information Services (IIS), Microsoft SQL Server, SQL Server Reporting Services, SQL Server Analysis Services, and Microsoft SharePoint. Although Service Manager planning and deployment guides make installation straightforward, configuring SQL Server, defining views, and configuring workflows and SharePoint is more complicated. However, these tasks are no less important since improper configuration of any of these components can lead to performance issues. In fact, analysts who use Service Manager often criticize the console's sluggishness. For example, customers who use Service Manager commonly complain about having to wait for a window to close and about seeing messages like "This item cannot be updated because it has been changed by another user or process." Performance issues like these can be frustrating, especially because the console is the only workspace analysts can use to manage service requests (apart from using third-party web-based consoles). Fortunately, most of these issues can be resolved through Service Manager customizations or by the analysts simply changing the way they use the console.

The official sizing guidelines for Service Manager (available from <http://www.microsoft.com/en-us/download/details.aspx?id=13605>) indicate that Service Manager can support up to 50,000 users and computers. This is not a hard limit, however. It simply reflects the scenarios that the product group has tested and documented. Several large customers with environments far beyond 50,000 users successfully use Service Manager. A core SQL Server infrastructure configuration with a focus on performance and optimization, together with a well-planned configuration of Service Manager itself, make it possible to scale well beyond 50,000 users.

This chapter provides guidance on optimizing the Service Manager environment for better performance of both the console and backend components such as the SQL database, workflow, and Portal. The chapter addresses some common mistakes and provides recommendations for avoiding pitfalls. These guidelines are mainly targeted for large environments (over 20,000 users), but some of the recommendations apply to all organizations regardless of size.

Service Manager management server

The Service Manager management server houses the main software portion of a Service Manager deployment. The first Service Manager management server to be installed acts as the workflow server and is critical to the environment. This role can be moved or failed over for disaster recovery. The following guidance applies mostly to larger environments:

- If supporting more than 40 to 50 Service Manager consoles simultaneously, add one or more dedicated Service Manager management servers and place these in a network load balancing (NLB) cluster. This will isolate the workflow management server and ensure the workflows are executed regardless of the number of users utilizing the Service Manager console.
- The most important resources to the management servers are RAM and CPU. Make sure to monitor the servers to ensure they are sized adequately, following the recommended specifications found in the official sizing guidelines.
- Always try to avoid adding other roles to the management servers, for example SharePoint or SQL Server. This is because the management server is the primary engine supporting the Service Manager infrastructure.

Service Manager console

The Service Manager console is the way that analysts interact with Service Manager. The perception they have concerning how Service Manager performs is therefore based on their experience when using the console. The following guidance is important to consider:

- Never maximize the Service Manager console because it impacts performance due to a Windows Presentation Foundation (WPF) issue. Instead, minimize the console and extend the window as needed, even to fill the entire screen; just don't use the maximize button. This is a one-time task since the Service Manager console size is stored and reused at next launch.
- If the Service Manager console is left running for a long period, for example overnight or longer, it can consume a large amount of memory. Because the Service Manager console is written in C# (a managed code language) it relies on the .NET Framework garbage collector to manage memory. Therefore, when there is available memory on the machine, the console will cache data in memory to improve performance. As memory demand increases from the operating system or other applications, the garbage collector will release the least amount of memory needed for the Service Manager console to function. Do not worry, therefore, about whether

the console is consuming too much memory; it will be released when other applications require it. For more information on this issue, see <http://msdn.microsoft.com/en-us/library/0xy59wtx.aspx>.

- Each view the user selects when using the console increases the console start time because every view is cached. For example, selecting the All Incidents view has a huge pull on the database, and therefore it takes a long time to load. Instead of many views with similar criteria, encourage users to use the filter and search functions to find the objects they are interested in. You can also chose which views will be available to each role when you configure user roles. And if a view is not needed at all, you can delete it from the General view.
- When creating views, use the smallest type projection possible. (If you're not sure what *type projection* means, see <http://blogs.msdn.com/b/jakuboleksy/archive/2009/01/20/getting-started-with-type-projections.aspx>). The following classes exist for Incident view:
 - Incident
 - Incident (Typical)
 - Incident (Portal)
 - Incident (Advanced)
- Depending on the class chosen, more components may be included in the view. The Incident (Advanced) type projection is the most inclusive, but it is also the most performance impacting class. Views based on the Incident (Advanced) class will thus have an impact on the console performance. As a result, you should always use the smallest projection possible. For more information, see <http://blogs.technet.com/b/servicemanager/archive/2010/12/02/faq-why-is-my-custom-incident-view-so-slow.aspx> where you will also find a link to a management pack with many useful type projections.
- Consider using the Global Operators group when assigning tickets to limit the number of users loaded when searching for anything assigned to Analyst. Doing this can result in a drastic performance improvement. (See <http://blogs.technet.com/b/servicemanager/archive/2012/12/14/faq-why-does-it-take-so-long-to-find-users-in-the-assigned-to-and-primary-owner-fields.aspx>.)

Service Manager databases

A Service Manager environment consists of two database roles: the Configuration Management Database (CMDB) and the Data Warehouse. The CMDB is a centralized, model-based repository for all service management information managed by the Service Manager server. This database contains three types of information:

- Configuration items (CI) such as knowledge articles and components from organizations imported via the Service Manager connectors
- Work items (WI) such as incidents, change requests, problem records, and so on
- Configuration settings for the product itself

The CMDB is where the console retrieves data to display in the views. It's also where changes or updates to any work item are written. The database is therefore very I/O intensive and is a critical component when it comes to performance. Follow this guidance for better CMDB performance:

- For optimal performance, place the Log database, TempDB database, and the Service Manager database on separate high performance logical unit numbers (LUNs) to optimize I/O.
- Configure the SQL Service to use 1 to 2 GB less than the total RAM available in the server to ensure the SQL Server does not exhaust the operating system.
- Use the official Service Manager sizing tool (<http://go.microsoft.com/fwlink/?LinkId=232378>) to estimate the database size, then configure the database for the expected size and disable autogrow. This will ensure the database isn't extended during normal working hours; instead, SQL administrators can manually extend the database if needed.
- The TempDB performance is critical to Service Manager performance. Use Multiple TempDB files, for example one per two cores (this is not necessary for the other databases). For more information, see [http://technet.microsoft.com/en-us/library/ms175527\(v=SQL.105\).aspx](http://technet.microsoft.com/en-us/library/ms175527(v=SQL.105).aspx).
- The default retention of work items in the CMDB is 90 days for incidents and 365 days for the others (change, release, service, and problem). You can decrease these settings and still comply with the requirement to have enough work items available in the console views. Decreasing the retention keeps the CMDB to a reasonable size. Remember that work items that are closed (not resolved, completed, and so on) are removed from the CMDB, but they are still available in the Data Warehouse for reporting purposes. For more information, see <http://blogs.technet.com/b/servicemanager/archive/2009/09/18/data-retention-policies-aka-grooming-in-the-servicemanager-database.aspx>.

The Data Warehouse acts as the long-term repository for the service management information collected by Service Manager. It can also be configured to store a subset of information from System Center 2012 Operations Manager and System Center 2012 Configuration Manager. When the Data Warehouse management group is installed and all features are enabled, seven databases are created:

- DWDataMart
- DWStagingAndConfig
- DWRepository

- DWASDataBase
- OMDWDataMart
- CMDWDataMart
- ReportServer

A Data Warehouse management group for a Service Manager environment is not a requirement and provides only reporting and long-term storage. If you are working in an environment where reporting is not needed or required, you can safely leave out the Data Warehouse. Not installing the Data Warehouse won't affect the usability of either the Service Manager console or the Service Manager Portal. If you plan to use reporting at a later stage, install the Data Warehouse at the start, not only to avoid losing needed data before reporting as required, but also to avoid impacting the performance of the Service Manager environment when too much historical data is synchronized at one time.

Some configuration changes can be made to improve the performance of the Data Warehouse and reporting. Consider these recommendations if the Data Warehouse is having performance problems, which is often the result when synchronization jobs from the CMDB fail. It's important to remember that the performance of the Data Warehouse has no impact on the console user's performance experience.

- Split some of the Data Warehouse databases across more SQL servers. However, keep in mind:
 - The DWStagingAndConfig and DWRepository databases must be on the same SQL Server instance.
 - The DWDataMart, OMDWDataMart, and CMDWDataMart can each be placed on different SQL Server instances. (There is no reason to place these databases on different SQL Server instances because the quantity of data and access to the data stored in the OMDWDataMart and CMDWDataMart databases are not large.)
- Place the SQL Server Analysis Server instance on a different SQL instance than the Data Warehouse databases. This will ensure that cube processing does not affect the extract, transform, and load jobs or the execution of reports. Processing of cubes is handled differently in SQL Server Standard than in SQL Server Enterprise; Enterprise can do incremental processing instead of just full. With SQL Server Enterprise, you can move the Analysis database to another drive. This is not an option in SQL Server Standard. If possible, use the Enterprise edition for the Data Warehouse. See the table in the next section for more information on SQL Server editions.
- By default, data in the Data Warehouse is retained for three years. You can modify this retention setting, but it requires some complicated changes. For more information, see <http://blogs.technet.com/b/servicemanager/archive/2011/06/07/how-much-data-do-we-retain-in-the-service-manager-data-warehouse.aspx>.

SQL Server editions

SQL Server 2012 and SQL Server 2008 are available in both Standard and Enterprise editions. Service Manager will function with either edition, but SQL Server Enterprise includes additional features that enhance your experience with the Service Manager Data Warehouse. For the Service Manager Database (CMDB), however, the Enterprise edition of SQL is not needed.

The following table lists the key differences between SQL Server Enterprise and SQL Server Standard for Service Manager environments. Carefully consider which SQL Server edition is best for your deployment.

ITEM	STANDARD EDITION	ENTERPRISE EDITION
Analysis Services Files	The Analysis Services database files are automatically placed in the default location.	You decide where Analysis Services database files will be placed.
Cube Processing	The entire cube is processed each night, and therefore the processing time increases as more data is accumulated. Cubes can still be queried during processing, but reporting performance will be reduced.	Cubes are processed incrementally each night, and therefore the processing time is reduced.
Measure Group Partitions	Measure groups are partitioned into a single partition thereby increasing the time it takes to process the partition.	Measure groups are partitioned on a monthly basis thereby decreasing the time it takes to process the partition.
PowerPivot	You can't use Microsoft SQL Server PowerPivot for SharePoint.	You can use Microsoft SQL Server PowerPivot for SharePoint.

Workflows

Incorrectly configured workflows heavily impact customer environment performance. If a workflow runs too frequently, performs large queries or bulk updates to work items, and so on, performance will suffer. Some Service Manager environments have thousands of workflows, resulting in slow and unreliable performance.

As described in Chapter 3, sometimes the processes in an organization need to be redesigned to align with Service Manager. Many times, customers want Service Manager to look and behave exactly like their current service management tool and do not consider it from a people/process perspective. Configuring Service Manager to mimic the incumbent tool usually results in installations with too many workflows, and a poorly performing Service Manager environment typically results. Therefore, to get the best out of Service Manager, you might need to redesign some of your processes to better align them with the tool.

Service Manager Self-Service Portal

The Service Manager Self-Service Portal, shown in Figure 6-1, provides end-user self-service and activity management. Service Manager Portal is based on SharePoint Server and is where end users can view the published service catalog, create service and incident requests, search for knowledge articles, and view, update, and approve review activities.

Deploying Service Manager Portal for end users usually reduces the number of calls to your service desk. This means your service desk has more time for other tasks such as incident and request management.

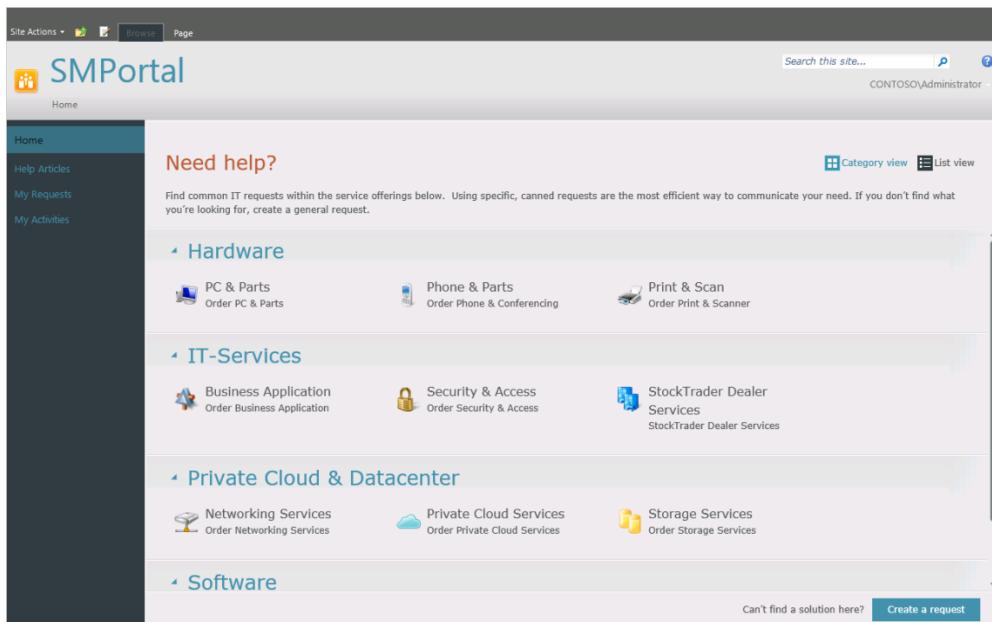


FIGURE 6-1 Service Manager Self-Service Portal

The following are some best practices for planning a deployment of Service Manager Portal:

- Service Manager Self-Service Portal for System Center 2012, 2012 SP1, and 2012 R2 is supported only on SharePoint 2010. SharePoint 2010 is supported to run only on Windows Server 2008 or 2008 R2, not on Windows Server 2012. Service Manager Portal is the only Service Manager component that cannot run on Windows Server 2012.

NOTE At the time of the writing, the product group in charge of System Center Service Manager was investigating whether to support SharePoint Foundation 2010 SP2 and whether to place the Self-Service Portal on Windows Server 2012. Until this is supported, the Self-Service Portal is supported only on SharePoint Foundation 2010 SP1 and Windows Server 2008 or Windows Server 2008 R2.

- Self-Service Portal can be installed in an existing SharePoint 2010 farm. If you're not running SharePoint 2010 or if you have already deployed SharePoint 2013, you can install Service Manager Portal on the free SharePoint Foundation 2010 edition.

NOTE For test purposes, it is possible to install the portal on a SharePoint 2013 installation. For more information see the blog post at <http://blogs.technet.com/b/thomase/archive/2013/08/19/how-to-get-service-manager-2012-self-service-portal-running-on-sharepoint-2013.aspx>.

- When the Web Content Server role is installed, make sure to verify that Application Pool Recycling is disabled. The default is nightly recycling, which results in a slow initial load for the users first accessing Portal. Also make sure you do not set recycling for high memory usage. For more information, see <http://blogs.technet.com/b/servicemanager/archive/2011/05/11/faq-why-is-the-self-service-Portal-so-slow.aspx>.
- When designing an environment that needs to support more than 20,000 users, you should add more Web Content servers and place them in an NLB cluster. This will ensure better performance when the Web Content servers are reading or writing data to or from the CMDB.
- When designing request offerings, aim to limit the number of custom icons used. Too many custom icons can have an impact on the time it takes to load the Portal main page.

Connectors

Connectors allow Service Manager to communicate with other products and technologies such as Active Directory and other System Center products. Connectors also allow Service Manager to import and store Configuration Items (CIs) in the CMDB. Using multiple connectors allows you to deploy a complex CMDB for the infrastructure. The following connectors are included out of the box in Service Manager:

- Active Directory connector
- Configuration Manager connector
- Operations Manager Configuration Item connector

- Operations Manager Alert connector
- Virtual Machine Manager connector
- Orchestrator connector
- Exchange connector (supported on Service Manager 2012 SP1 and later.)

General considerations

Some general considerations concerning connectors include:

- For both out-of-the-box and third-party connectors, always use a unique run-as account for each connector. Doing this will create a separate MonitoringHost.exe process on the workflow management server for each connector running on the server, making it easier to see which connector is currently running and how much memory/CPU it is consuming. It also makes it easier to isolate one connector from other connectors so that it can be terminated without affecting other connectors or workflows.
- Configure your connectors to avoid having them pull large amounts of data into Service Manager during the day, thereby affecting performance. All of the connectors are configured via the Administration tab in the console, except the Active Directory and Virtual Machine Manager connectors, which are configured using management packs.

Active Directory connector

The Active Directory connector is used to synchronize data between Active Directory Domain Services (AD DS) and the Service Manager CMDB. Service Manager can synchronize the following types of directory objects:

- Users
- Computers
- Printers
- Groups

Customers often configure Active Directory connectors to import all objects from the root of the domain. However, this means that all users (enabled and disabled), groups, and computer objects are imported, causing the Service Manager database to bloat with objects that are not needed. So, the first thing to consider when configuring the Active Directory connector is whether you need all directory objects in the CMDB.

Computers

If you are also using the Configuration Manager connector, there really isn't a need for the Active Directory connector to import all computer objects since it would only mean that

Service Manager would have to import, merge, and maintain two sources. All relevant information about computers in the environment is delivered by the Configuration Manager connector.

Groups

Work items are often assigned to support groups first and then to individual user accounts, so they don't contain membership information. Therefore, groups are often not used in Service Manager. So instead of importing all Active Directory groups, filter to import only the relevant groups as needed.

Users

The Active Directory connector imports all users in a domain, regardless of whether they are enabled or disabled. If contacts are created as domain users in the directory, these are imported as well. It is therefore important to consider which organizational units (OUs) to import users from and also whether to import both enabled and disabled users. Typically, depending on your OU structure, it is best to create separate Active Directory connectors to avoid populating the CMDB with unnecessary data.

For example, if you want to configure a connector to import only enabled user accounts, you can use the LDAP filters that were introduced in Service Manager 2012. Launch the Active Directory connector wizard and, at the Select Objects option shown in Figure 6-2, select Users Or User Groups and enter the following LDAP query:

```
(&(ObjectCategory=User) (! (userAccountControl:1.2.840.113556.1.4.803:=2)))
```

This LDAP filter will ensure only enabled user accounts are imported.

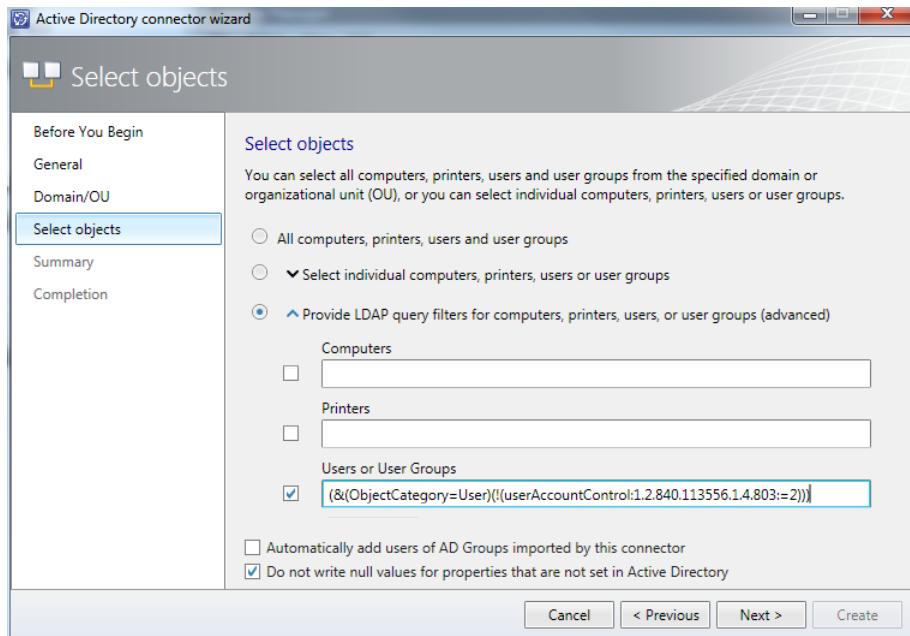


FIGURE 6-2 The Active Directory Connector Wizard

NOTE Always select the Do Not Write Null Values For Properties That Are Not Set In Active Directory option to ensure the connector does not update the same attributes despite being null. If you need to write null values from Active Directory, however, do not select this option.

The following are some best practices for designing the Active Directory connector:

- For the most optimal planning and performance of the Active Directory connector, first perform a thorough cleanup of any obsolete objects in the directory and identify which OUs to import into the CMDB. Some environments include thousands of obsolete objects that can be mistakenly imported into the CMDB. Having large numbers of obsolete objects in the CMDB will impact console performance since some user information is cached in the console at startup. Importing obsolete objects will also impact Service Manager performance. In addition, obsolete and irrelevant user objects will impact analysts during a user lookup.
- For the Active Directory connector, decide which directory objects need to be part of the CMDB. Set up one or more Active Directory connectors that import only relevant data. A poorly managed directory can contain thousands of stale objects that aren't needed in the CMDB.
- When designing the Active Directory connector, consider using LDAP filters to import

- only relevant types of objects. For more information on LDAP filters, see [http://msdn.microsoft.com/en-us/library/windows/desktop/aa746475\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa746475(v=vs.85).aspx).
- If your design includes several Active Directory connectors that daily import large numbers of changes, consider scheduling the time each connector runs, taking into account the duration that each connector needs to complete, so no more than one Active Directory connector runs at a time.

Operations Manager connector

There are two different Operations Manager connectors. Operations Manager *Configuration Items* (CI) connector is used to synchronize CI data from Operations Manager to the Service Manager CMDB. Operations Manager *Alert Connector* is used to synchronize Alerts to and from Service Manager. The Operations Manager Alert connector is a two-way connector, meaning that when an incident related to an Operations Manager alert is resolved, the Alert connector will also close the same alert in Operations Manager.

The following are some best practices for designing the Operations Manager Alert and CI connectors:

- If you plan to use the Operations Manager Alert connector to have alerts automatically created as incidents in Service Manager, do not create the connector until you have tuned the Operations Manager environment to report only relevant actionable alerts. Some customer environments forward all alerts from Operations Manager to Service Manager despite many of them being false positives. Besides affecting the performance of Service Manager, this also affects the usefulness and trustworthiness of Service Manager since not all incidents need to be acted upon.
- Begin with only a few management packs from Operations Manager to import alerts into Service Manager. Create incident management templates with predefined values for Support Group and Incident classification per management pack to start off slowly. Export all the management packs in Operations Manager and provide this list to the IT infrastructure team and ask: Who needs this alert assigned to them? Then create incident templates using the information that IT provides to streamline the assignment of alerts to the groups that are responsible for them.
- Be very selective concerning which Operations Manager monitored objects need to be added to the CMDB. There is no requirement for importing all Operations Manager monitored objects into the CMDB. The more objects that are imported into the CMDB, the more objects need to be maintained by the connectors, workflows, and so on. For more information, see <http://blogs.technet.com/b/servicemanager/archive/2010/02/26/managing-the-allowed-list-for-the-operations-manager-ci-connector-with-powershell.aspx>.

Configuration Manager connector

The Configuration Manager connector is used to synchronize data between the Configuration

Manager database and the Service Manager CMDB. Service Manager synchronizes hardware and software inventory as well as asset intelligence inventory attributes. In addition, Service Manager provides a workflow for creating incidents when managed machines are reported as non-compliant by using the Desired Configuration Management (DCM) feature in Configuration Manager.

The following are some best practices for designing the Configuration Manager connector:

- If you are not using DCM in your environment, you should disable the DCM rule in Service Manager. For more information, see <http://blogs.technet.com/b/mihai/archive/2012/11/30/configuration-manager-connector-s-dcm-rule-can-cause-massive-performance-issues-in-service-manager.aspx>.
- Always consider which collections in Configuration Manager to import because selecting All Systems might import data from systems that aren't relevant to store in the CMDB. See Figure 6-3 for more information.

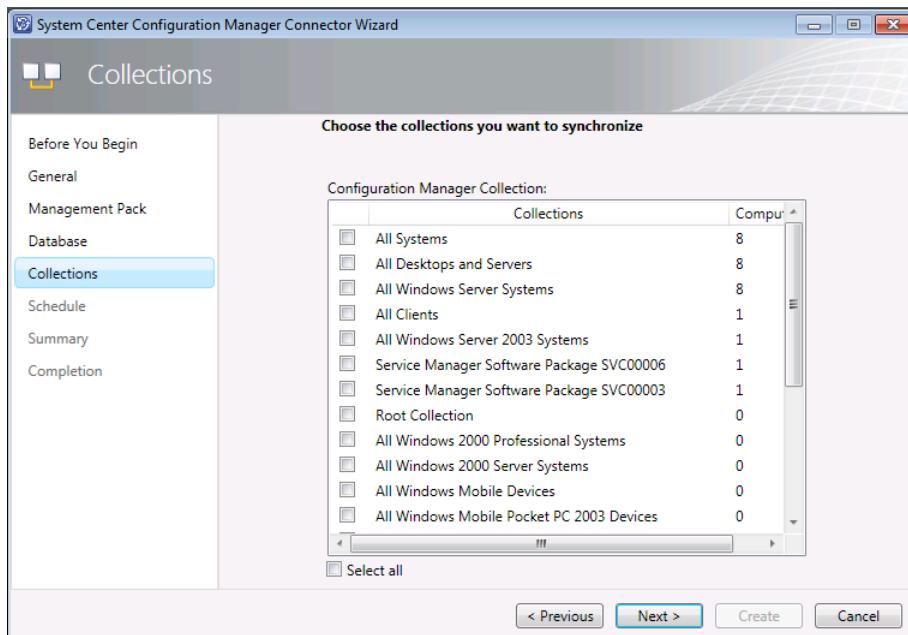


FIGURE 6-3 The Configuration Manager Connector Wizard

Orchestrator connector

With the Orchestrator connector, you can synchronously invoke Orchestrator runbooks from within Service Manager by using workflows. This capability integrates Orchestrator automation capabilities and the Service Manager Portal, as well as business modeling capabilities.

Activities that make up a service request can be mapped to runbook activities, which in turn are mapped to an Orchestrator runbook. For example, the parameters that are necessary for a custom start activity to invoke a runbook in Orchestrator, such as a computer name, can go into Service Manager as objects. You start this process by importing runbook objects into the Service Manager database using the Orchestrator connector.

If you expect to run many Orchestrator runbooks that will involve Service Manager and you have identified that this might degrade the performance of your environment, add an additional management server and target all Service Manager-related runbooks to that server. This server's sole role will then be to reply to Orchestrator requests in isolation from other Service Manager workflows or to support console access, which is housed on other management servers.

Service Manager configuration and customization

While Microsoft System Center 2012 Service Manager is a complete IT Service Management (ITSM) solution and many customers have successfully implemented it "out of the box," some situations call for additional configuration and customization. Service Manager can be configured through the console or customized externally to meet customer requirements.

One of Microsoft's technical principles is "configure before customize," which leads one to ask: Why is there a section in this book on customizing Service Manager when Microsoft recommends configuration over customization? The answer is simple: The economy has changed and customers expect shorter value runways from IT solutions. Historically, IT has generally preferred customizing purchased systems before even understanding whether existing functionality could be leveraged to meet their needs. In this scenario, a lack of training and product knowledge frequently results in unnecessary customization as well as potentially problematic misconfiguration.

Existing out-of-the-box capabilities in Service Manager should be understood and exploited before customizing a solution. As you will see in this chapter, however, some areas of functionality specifically require customization. This chapter provides "safe" customizations and configuration recommendations that have been accomplished with or by Microsoft partners and customers.

The good news is that both customizations and configurations are stored in management packs as described in previous chapters, and for the most part these management packs "survive" upgrades. While such survivability is intended as part of the Service Manager architecture, upgradeability is not fully guaranteed. For example, if any reserved keywords used in a custom management pack by a customer are used later by the Microsoft product group, there could be a conflict.

The configurations and customizations described in this chapter are not required by any means, and should be implemented only when an actual justified business need is identified. They have all been implemented previously by Microsoft Services and partner consultants for Service Manager customers in order to address their stated requirements.

Configurations

Configurations are made through the Administration or Library workspaces within the Service Manager console and can be performed using either the Administrator or Author security role. While configurations are generally not laborious or complex, they can still create issues with the IT organization through inefficiencies and misunderstanding of values and usage. The following configurations are described in the sections below:

- Incident and service request support groups and assignment
- Incident categorization

Incident and service request support groups and assignment

Assignment of incidents and service requests is a "gap" in Service Manager. A traditional ticketing system typically includes a group selection and then individuals within that group selection. Since Service Manager doesn't provide inter-field dependency, there is no connection between the enumerated list "Support Group" and the user/group relationship "Assigned To." This must therefore be implemented manually or using a template setting because if queues have been defined against support groups, an inaccessible incident or service request may be assigned.

Problem

Support Group and assignment of incident and service request are not related to one another.

Solution

Kurt Van Hoecke's solution on the SCUG.BE blog provides the ability to assign incidents and service requests directly to an analyst based on membership in a support group that aligns to an Active Directory security group set in the library settings. You can find the solution at <http://scug.be/scsm/2013/06/23/scsm-incident-assignment-add-on/>. A similar solution is commercially available from Cireson and is available from <http://www.cireson.com/app-store/scsm-group-assign/>.

Incident categorization

Incident classification category values must be modified in order to avoid duplicate configuration item data and to drive proper reporting/dashboards. Because these values are presented as an enumerated list in Service Manager, it is easy to modify this item. However, it is one of the most important values in incident management and also one of the most hotly debated.

Problem

The out-of-the-box classification category values included in Service Manager are an illustrative list of values meant to demonstrate the capabilities of the product. These out-of-the-box values are:

- E-Mail Problems
- Networking Problems
- Printing Problems
- Software Problems
- Hardware Problems
- Configuration Data Problems
- Enterprise Application Problems
- Other Problems

Category values can be a politically charged discussion because of their intended use and governance issues. In legacy ITSM solutions, category was often overloaded with terms like configuration item, service, problem (misused), organization, and even resolution. This was the world of ticketing. But with the onset of ITIL, ITSM best practices, and modern software development, most modern ITSM solutions use:

- *Configuration item* for the exact item or items the incident is about
- *Service* for the exact service or services the incident is about
- Resolution codes for how the incident was resolved

What still remains, however, is to use a classification category to describe the incident itself rather than what it is about.

Solution

The classification category should describe the incident itself rather than what it is about. And if true self-service is a goal when implementing Service Manager, as it should be, classification category should be expressed in terms businesses and customers can understand. Simplicity is the key here. The following are some recommendations for incident classification:

- Keep the number of values to a minimum. A common rule is 10 x 10, which means no more than 10 parents with 10 children each. This helps facilitate reporting along classification category values.
- Eliminate *other* as an option.
- In non-IT terms, follow the KISS Principle.
- Minimize opportunities for overlap (e.g., a printer problem could be a hardware problem).
- Do not include the term *problem* in the value unless it is absolutely asked for by

people outside of IT. *Problem* as a term has specific meaning in problem management.

- Classify the incident, not what it is about (that's what *configuration item* and *service* are for).
- The process should seek to eliminate the need to change it more than once (a performance category should apply to both configuration item and service). If it is changed more than once, consider this a process failure.

Below are some recommended configurations in this area. These configurations have worked well for some organizations and can be modified and extended for your own.

- **Access** Parent container and general value for access-related incidents
 - **Access Denied** Incidents as a result of Access Denied message
 - **Permissions** Indication that user doesn't have proper permissions
 - **Password Problem** Password-related errors (not including password reset)
 - **Login Failed** User and system-related login failures
- **Fault/Failure** Parent container and general value for faults or failures
 - **Alert** Used for System Center Operations Manager alerts
 - **No Power** Power-related issues
 - **Feature Failure** Failure of a feature of service
 - **No Response** No response issues
 - **Communication** Communication issues
 - **Frozen** Frozen service
- **Performance** Parent container and general value for performance-related issues
 - **Response Time** Response time
 - **Outage** Outages (can be confused with Failure)
- **Data** Parent container and general value for data-related issues
 - **Lost Data** Missing or lost data
 - **Quality** Quality of data issues

Customizations

Customizations take place outside of the Service Manager console and involve something being created and/or imported externally to the console. Customizations are almost always XML based using the Service Manager Authoring Tool, Visual Studio, Blend for Visual Studio, or other XML editors and development tools. They can also come from community-contributed or partner-provided management packs. The following customizations are

described in the sections below:

- Work item custom labeled fields
- Notification Boolean
- Assignment notification for all work items
- More information needed and possible problem
- Location on form
- On Behalf Of on form
- VIP incidents and requests
- Change requests submitted from the self-service portal
- Change phase in change views

Work item custom labeled fields

When creating Service Manager request offerings, while defining the request, you must map user inputs from the Self-Service Portal to properties and relationships in the incident, service request, or child activities associated with the request offering. The challenge comes when there are more user inputs in the request offering than properties and relationships to support them in the related template.

Problem

The out-of-the-box work item classes (incident, request, change, and activity) have predefined fields and relationships. They do not have custom or generic properties, nor is there a means of displaying them for mapping request offering prompts.

Solution

Extend work item classes (particularly service request and incident, and also activity if necessary) with generic custom fields and text labels so the label can indicate that the property is purposed.

Customize the work item form with a custom fields tab that includes labels. Since the fields can be purposed differently, however, they should not be included in the data warehouse for reporting.

You can extend work item classes using the data types shown in Table 7-1. The custom field has the specified data type shown while the label is always a string.

TABLE 7-1 Properties and data types for extending work item classes for custom labeled fields

PROPERTY	DATA TYPE
CustomTxt1	String
CustomTxt1.Lbl	String
CustomTxt2	String
CustomTxt2.Lbl	String
CustomTxt3	String
CustomTxt3.Lbl	String
CustomTxt4	String
CustomTxt4.Lbl	String
CustomDate1	Date/Time
CustomDate1.Lbl	String
CustomDate2	Date/Time
CustomDate2.Lbl	String
CustomBool1	Boolean
CustomBool1.Lbl	String
CustomBool2	Boolean
CustomBool2.Lbl	String
CustomInteger1	Integer
CustomInteger1.Lbl	String
CustomInteger2	Integer
CustomInteger2.Lbl	String
CustomReal1	Real
CustomReal1.Lbl	String
CustomReal2	Real
CustomReal2.Lbl	String

Customize the respective form and create a new tab on the work item (Service Request Form is used in the example shown in Figure 7-1) and position all of the fields in a table. Place the text label strings (Lbl) above the custom fields. Remember that the authoring tool records all updates to the form and "replays" them when the form is rendered.

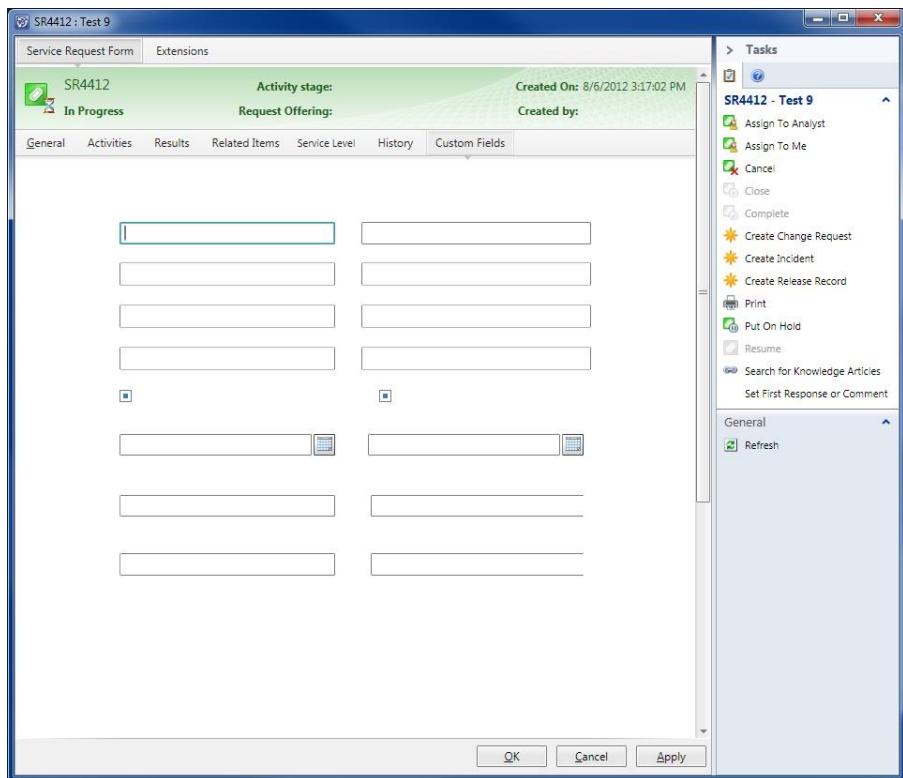


FIGURE 7-1 A custom fields tab of a service request.

When creating templates, you can pre-populate the custom label fields by using the Extensions tab at the top of the template form, as shown in Figure 7-2.

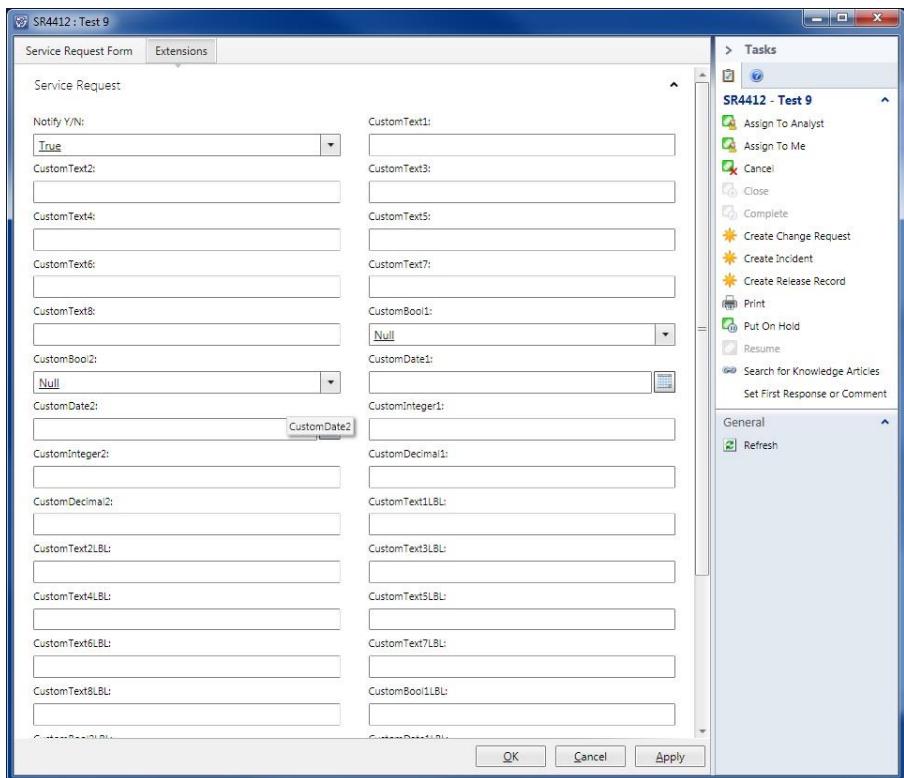


FIGURE 7-2 The Extensions tab of a service request.

When an instance of the work item is viewed by a user from the console, the fields with labels are used and the ones without labels are not. These properties can then be used by workflows in Orchestrator and in notifications.

Notification Boolean

Service Manager can send an email message on virtually any condition or trigger in the system. However, in certain scenarios, you might want to turn notification on or off with a template (as a default value) instead of enabling or disabling notifications. You might also want to do this on a per-work-item basis.

Problem

You need to be able to turn off notification on a per-template or even per-work-item basis.

Solution

Extend the work item with the new property `NotifyYN`. This property has a Boolean data type and the default value should be set to `True`.

Next customize the form to include "Notify Y/N" as shown in Figure 7-3. Include NotifyYN equals True in all notification criteria, including any management packs that provide notification from external providers.

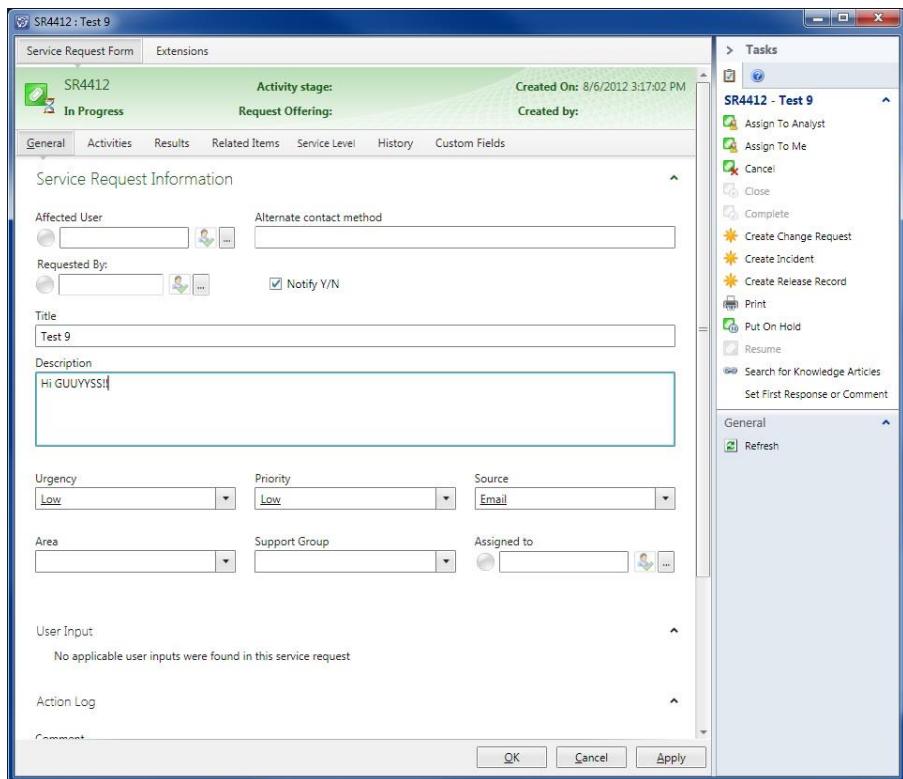


FIGURE 7-3 A service request with a modified form.

Assignment notification for all work items

Configuring subscriptions in Service Manager provides the ability to send notifications based on specified criteria. Unfortunately there is no means of triggering notification based on any assignment change from the console. Non-specific subscriptions are often desired for assignment change notifications of incidents, service requests, changes, and activities so that a notification subscription for each assignment group is not needed. Defining discrete assignment notifications per assignment group is not recommended because that could become a performance issue in the workflow engine.

Problem

There is no means to configure modular notifications based on assignment change, and a large number of discrete notification workflows can cause performance issues.

Solution

There are a number of commercially available solutions for this problem as well as some custom solutions described in the Service Manager blog on TechNet. For example, there is a free and licensed version of notification from Microsoft Partner Cireson at <http://www.cireson.com/app-store/>. The only issue with this is that if other logic, such as NotifyYN, is applied to notifications on assignment, it won't be checked. However, this is not common with assignment notifications.

Several easily implemented, open assignment notifications have been published in TechNet blog posts. Specifically, a blog post for incidents can be found at <http://blogs.technet.com/b/servicemanager/archive/2009/12/15/custom-notification-workflow-on-incident-assignment-or-re-assignment.aspx?PageIndex=2>, and a blog post for activities can be found at <http://blogs.technet.com/b/servicemanager/archive/2010/03/12/custom-notification-workflow-on-activity-assignment-or-reassignment.aspx>.

The Microsoft community has also created notification management packs for Service Manager request and release work items (see <http://www.bctechnet.com/scsm-notification-workflow-on-work-item-assignments-part-2/>). Because these are unsealed management packs, you can add logic to the XML to test a Boolean such as NotifyYN or other logic.

More information needed and possible problem

Commonly, issues exist between different escalation levels of a service desk. All too often the service desk quickly transfers an incident to a second level resource, and the second level resource sends it back, indicating there isn't enough information in the incident or request to complete it.

Since this scenario indicates a process defect, there should be a capture of information any time an incident that hasn't adequately followed good first level resolution procedures is passed along to second or third level resources. Furthermore, to drive improvement of first level service desk personnel, there should be a capture of what information is required in order to escalate. This can be in the form of a text property in the incident or a related knowledge article.

To further complicate matters, there is usually no means by which to alert problem management that a possible problem exists based on the incidents that are coming into the service desk.

Problem

Other than the action log, there is no means for highlighting inadequate information sent from higher support tiers back to lower ones, nor any means to highlight the possible problem.

Solution

To address this problem, extend work item classes with the properties shown in Table 7-2.

TABLE 7-2 Properties and data types for extending work item classes when more information is needed and for identifying a possible problem

PROPERTY	DATA TYPE
Possible Problem	Boolean
Inadequate Information	Boolean
Inadequate Information	String (long text)

Customize the incident form on the Resolution tab to include Possible Problem, Inadequate Information, and Inadequate Information on the form.

Set up a trigger to notify the problem manager when an incident is saved with Possible Problem checked.

Set up a trigger to notify the incident manager when an incident is saved with Inadequate Information checked.

Establish a view of incidents for problem managers and analysts so they can see all active and resolved incidents where a possible problem has been flagged.

Location on form

An incident or service request frequently includes either editable properties for location or information about where one can find the person when servicing the incident or request. But in our mobile world, what is recorded in Active Directory may not always be accurate. In service management, an incident or service request therefore needs dynamic properties that contain the location information about the affected user and the particular work item.

Problem

The location needs to be overwritten for an incident or request when the customer (the affected user) is not at their normal location, for example in the case of a remote worker or traveling employee.

Solution

To address this problem, create properties that map from user information such as department, location, office, or other customer properties in Active Directory. This information should be bound to the properties in the work item in such a way that they are populated with Active Directory information but can also be modified if necessary. This requires a OneWay binding between the work item location property and the user location in Active Directory.

The following binding modes are available in the authoring tool for editing forms:

- **OneWay** This binding mode causes changes in the source property to automatically propagate to the target property, but changes to the target property are not propagated back to the source property. This type of binding is appropriate if the control being bound is implicitly read-only.
- **TwoWay** This binding mode causes changes to either the source or target property to automatically propagate to the other. This type of binding is appropriate for editable forms and other fully-interactive user interface scenarios.
- **OneWayToSource** This binding mode is the reverse of the OneWay mode and updates the source property when the target property changes. You might use this, for example, if you only need to re-evaluate the source value from the user interface.
- **OneTime** This binding mode causes the source property to initialize the target property, but any subsequent changes do not propagate. This means that if the data context undergoes a change or the object in the data context changes, then the change is not reflected in the target property. This type of binding is appropriate if the data is static or if you are using data where a snapshot of the current state is appropriate to use.

On Behalf Of on form

Customers often have users with administrative assistants, as well as situations when a user's computer is unable to access the Self-Service Portal and he or she wants to submit "on behalf of" another person. This is accomplished by leveraging the already existing relationship for Requested By in incidents and service requests and simply exposing it in the forms.

Problem

Need to add On Behalf Of for the affected user for incidents and service requests.

Solution

Incidents and service requests already have a relationship called Requested By User that can be used for this purpose. Alternatively, an additional relationship can be added to the class. The forms can then be modified to include another user picker as in the Requested By field in the form shown in Figure 7-4.

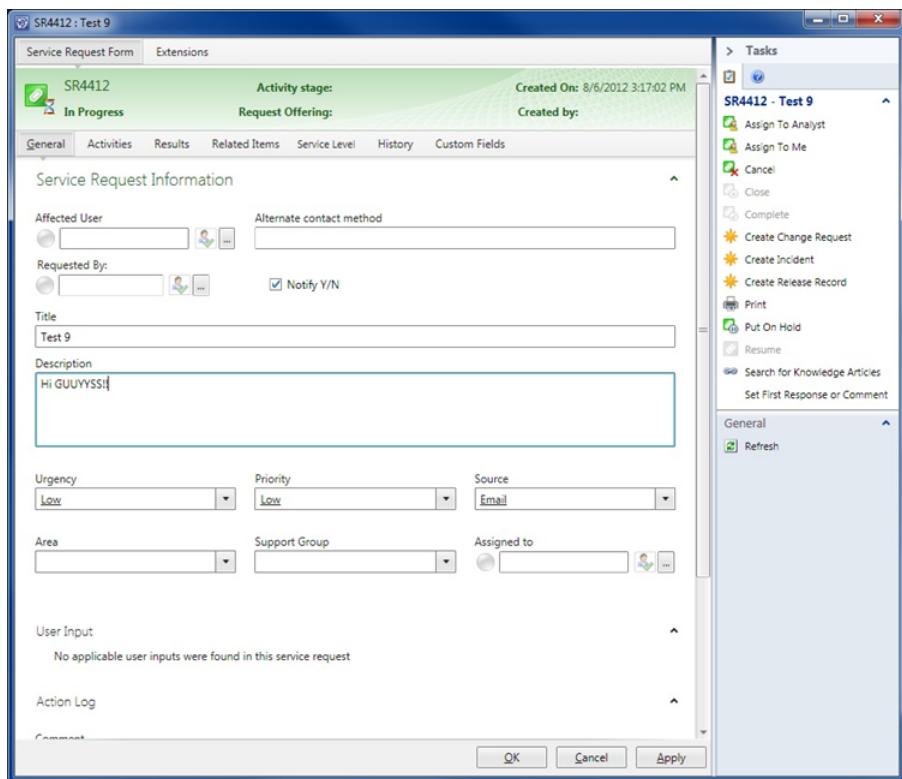


FIGURE 7-4 An example of a Requested By user picker.

This user relationship for Requested By can be included in any notifications along with the affected user. It will simply be ignored if the relationship doesn't exist.

VIP incidents and requests

A very common need in ITSM is identifying important individuals in leadership, representatives, and power users. Often they need to be given "VIP Status" because of who they are and why they are contacting the service desk.

An effective incident and service request process should not automatically escalate impact or urgency just because an individual is a VIP. These callers should, however, be given the option to indicate their VIP "management override" in order to get higher priority. Procedurally, they should be offered standard status based on impact and urgency, but given the opportunity to trump priority based on their VIP credentials.

Problem

Service Manager doesn't provide a means of identifying VIP users.

Solution

To address this issue, extend the User object with a text property for special status (this can be used for more than just VIP). Indicate "Special Status" or "VIP" in the text field.

Customize the incident or service request form to provide a label that is bound to the affected user special status. Use coloring such as red to highlight it on the screen.

Add a VIP value for Urgency or Impact, depending on how the organization wants to define terminology, and set priorities accordingly for the VIP row or column in the table.

Submitting change requests from the Self-Service Portal

Customers frequently request the ability to create change requests from the Service Manager Self-Service Portal. This capability is not provided out of the box, and one could argue that this is actually not a bad thing. Change management should strive for a high level of quality. That means proper intake, execution, and completion of changes. If this isn't taking place, change management is not working properly. The idea here is that service requests make a good change request intake mechanism for several reasons:

- The end user can be asked a series of change questions as a request offering, and information can be captured and used when the requested change is implemented.
- Changes that should have never been submitted can be eliminated via service requests, allowing for better optimization of the change management process.

Problem

Submitting change requests via the Self-Service Portal is not possible out of the box.

Solution

The solution requires at least one service request template and one change request template. System Center 2012 Orchestrator is also required.

The template shown in Figure 7-5 is a real-world example where a customer wanted IT managers to:

- Complete a review of a submitted change (as manual activities) to decide on a category, and add and update the change request and activities.
- Automatically proceed with the change request when two managers or at least 33 percent of all assigned approvers have approved the change.

The service request contains a dependent activity (DA) which keeps the service request "in progress" until the change is completed and then completes the dependent activity and finally the service request.

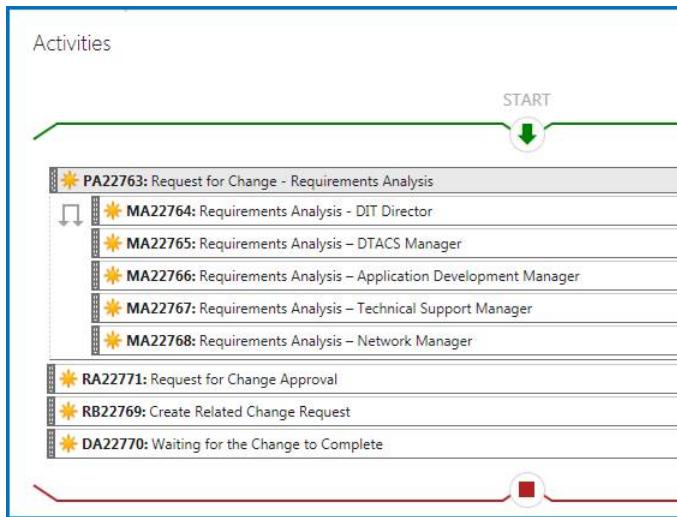


FIGURE 7-5 Service request template to create a change request using Orchestrator.

Note that the final activity in the Service Request is a dependent activity. This "holds" the service request in "in progress" and is completed or failed using a runbook from the related change.

As Figure 7-6 shows, the first runbook is triggered when the service request-based change request is approved and the appropriate change request is created. As the figure shows, different types of changes can be created from the contents of the service request.

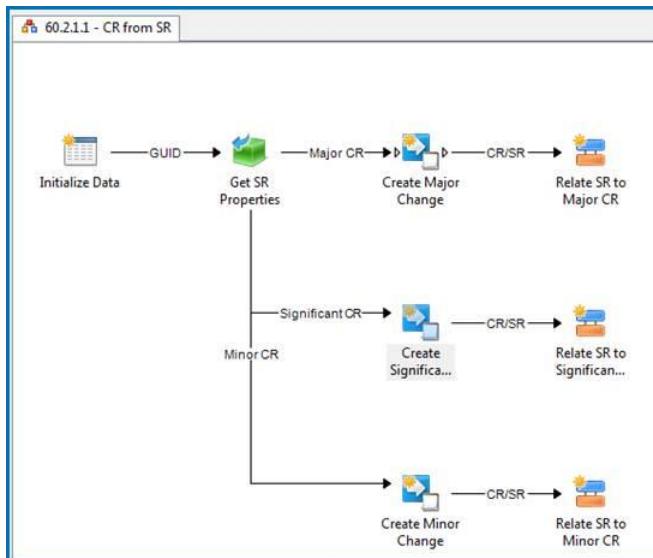


FIGURE 7-6 Runbook to create a change request from a service request.

The runbook also links the service request and change request (see Figure 7-7) so that when the change is completed, another runbook can be leveraged to complete the dependent activity in the service request (see Figure 7-8).

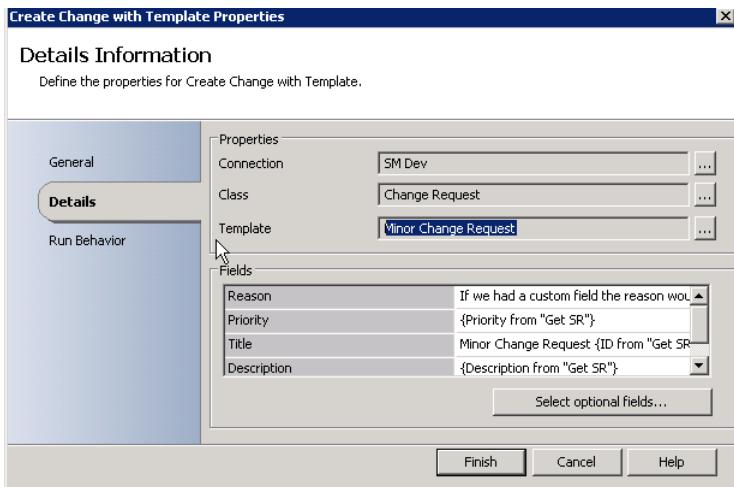


FIGURE 7-7 Runbook activity to create a change request.

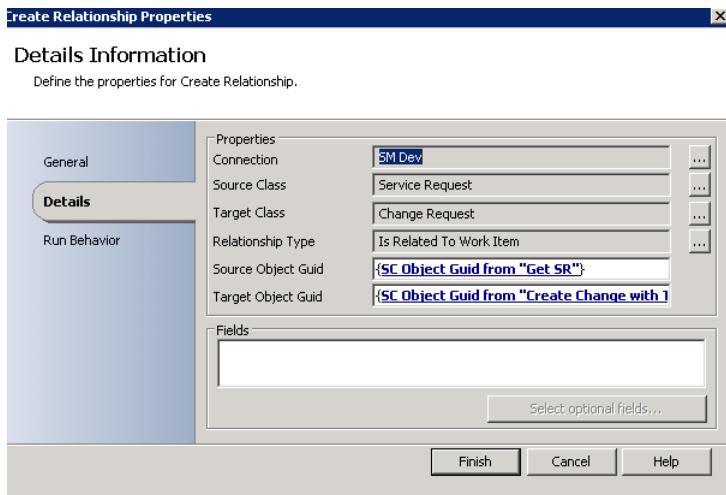


FIGURE 7-8 Runbook activity to create a relationship between a service request and a change request.

The change request template should include an appropriate set of activities and a runbook activity at the end that finds the related service request, finds the dependent activity, and completes it as shown in Figure 7-9.

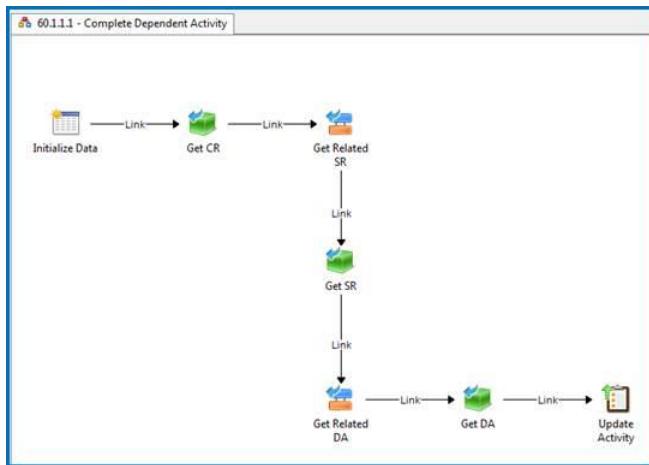


FIGURE 7-9 Runbook activity to complete a dependent activity in a related service request.

Change phase in change views

In one real-world customer scenario, a change manager wanted to view change requests grouped by phase or stage of the change. Change requests in Service Manager can be New, Submitted, In Progress, Failed, Cancelled, Completed, or Closed, and changing these status options is not advised since they are driven automatically by change management workflow.

Ideally, similar to service requests, change requests should go from New to In Progress to Complete to Closed. In reality however, change requests typically have a staged lifecycle and different levels of governance and review based on change category. Here are some examples of definitions taken from the Microsoft Operations Framework (MOF) and from the experience of the authors:

- **Standard change** This category is low risk because it has a proven set release path. It has minimal business impact and a known set of release procedures. Typically, standard changes do not include review activities for approval.
- **Minor change** This category typically affects only a small percentage of users and resources. Also, the risk of an outage is less because of the organization's experience in implementing this type of change. Minor changes are low-risk, low-scope changes that require only a minimal amount of management oversight. Minor changes should be targets for becoming Standard changes and usually require a single level of approval (e.g., Deployment CAB or Change Manager approval).
- **Significant change** This category has a moderate effect on users, resources, and the business. It might involve downtime of services and may also involve a situation in which the organization has less experience with the product, infrastructure, or the client involved in the change. Significant changes do not include business approval and are for high-risk, high-scope changes. Often, significant changes can be modeled

or created via a template (Microsoft calls these Known Change Types) and have multiple levels of approval (e.g., Technical CAB and Deployment CAB).

- **Major change** With both high risk and high cost, this category involves the greatest potential impact to users and resources. It might also affect a business-critical systems and could involve downtime of the service. Major changes involve the business (external to IT, or COOP, etc.) if services are being changed, added, or removed and have a material impact on business. These changes are typically minimally modeled or created via a template and have multiple levels of approval (e.g., Planning CAB, Technical CAB, and Deployment CAB).
- **Emergency change** This category is high risk because of the urgency of release and the minimal time available to test the change. It is relatively uncertain if the change will succeed, and there is a big impact on the business if it fails. This type of change is often a result of an urgent incident. These changes are escalated to the Emergency CAB for fast-track approval.
- **Unauthorized change** Unauthorized changes should be tracked when they are detected to report and minimize their occurrence. This level involves changes that occur outside of the agreed-to change management policies or changes that are specifically forbidden. Activities should focus on correcting the CMS or backing out the unauthorized change, depending on what makes business sense.

You can use the enumerated list values of the Stage property to identify phases by the activities of the change request. Out of the box, these values include Initiate, Approve, Develop, Test, Release, and Validate and Review. Since this is an enumerated list, these values can be modified, but it's important to note that they are used by all activities across all work items.

Problem

There is a need to group change requests into views based on the stage of the change request.

Solution

To address this need, confirm the enumeration list values for activity stages, then extend the Change Request property with a text field. For example, it might be called Change Request Stage.

Customize the change request form to view Change Request Stage as a label (non-editable).

Monitor completion of activities that are child activities of a change request, and if the value of Stage is set to something other than blank, set the Change Request Stage of the parent change request to the stage specified in the activity.

Create or modify a change request view to group changes by Change Request Stage.

Additional resources for configuration and customization

Many great sources offer numerous other configuration and customization options. The source identified in the previous sections are not so common and involve Microsoft recommendations for best use of Service Manager.

To find almost any customization management pack for Service Manager, simply use Bing or some other search engine to search for SCSM 2012 <functionality sought>. For example, a search for:

SCSM 2012 incident auto closure after five days

brings up as its first result an article titled "SCSM 2012: Auto Close resolved incidents after x days" from the System Center User Group Belgium website. Often, multiple possible solutions can be found in this way.

Also, a System Center 2012 Service Manager Survival Guide links to TechNet articles and to many blogs where you can find community-contributed management packs that close up gaps in the solution. You can find the Survival Guide at <http://social.technet.microsoft.com/wiki/contents/articles/8113.system-center-2012-service-manager-survival-guide.aspx>.

Customization risk areas

The customization of Service Manager is not without risk. Customizing or even configuring any process-centric system, such as Service Manager, can create a number of inherent risks that the guidance in this book is intended to minimize. These risks can include:

- Not using properties and relationships for their intended purposes, thereby invalidating some of the out-of-box workflow, reporting, and analysis capabilities of Service Manager (e.g., classification and support groups on incident management).
- Driving inefficient or risky work patterns in both human and system-centric processes.
- Making modifications that might limit future upgradeability and supportability of the solution.

One of the design criteria Microsoft had for Service Manager was to specifically address the third risk mentioned above, which is one of the industry's biggest issues, that is, allowing customer modifications to the functionality of the system in a way that won't limit the ability for the system to be upgraded and supported. By keeping customer configurations and customizations within management packs, Service Manager is able to preserve and separate customer specific settings, workflows, and notifications, from the underlying Service Manager engine provided by Microsoft.

However, neither Service Manager nor any other ITSM solution can completely mitigate the first two risks since these are design and implementation choices made by the customer. It's very important, therefore, to understand the desired outcome of a Service Manager implementation. Having knowledge of only a legacy ITSM tool is dangerous to take into a new Service Manager deployment.

Orchestrator versus Authoring Tool for workflows

Workflow can be accomplished in Service Manager through several different means:

- **Activities in Service Request, Change, and Release Management** Incident work items can have activities, but there is no capability to sequence them and any activities go into "In Progress" immediately.
- **Workflow Settings in Administration** These are work-item specific and you can change property and relationship values only through the application of templates. Notifications can be configured, but this should be done only when the notification is desired in conjunction with the template.
- **Service Manager Authoring Tool** This tool provides a graphic workflow design based on Windows Workflow Foundations (WWF). Most workflow configuration is done through .NET or Windows PowerShell scripts.
- **System Center Orchestrator** Formerly known as Opalis Runbook Automation, Orchestrator provides an Integration Pack for Service Manager.

Although using the Service Manager Authoring Tool for workflow is still supported through Service Manager 2012 SP1, with the acquisition of Opalis (Orchestrator), no further enhancements are planned for workflow in the Authoring Tool. Microsoft therefore recommends that Orchestrator always be implemented along with Service Manager. Because Orchestrator is a scalable solution, only a small footprint is needed for Service Manager-specific automation. Broad Orchestrator implementation can come later where the larger benefits of Orchestrator can be realized, which can be significant.

Some other benefits of using Orchestrator instead of authoring workflows using the Authoring Tool include:

- Orchestrator runbooks that leverage the Service Manager Integration Pack require less time and knowledge of Windows PowerShell and scripting.
- There is better error/fault handling in Orchestrator in case the runbook should fail.
- Authoring workflows using the Authoring Tool will result in a management pack and DLL file. The management pack must then be imported into Service Manager, and the DLL deployed to all management servers.

Below is an example of some Windows PowerShell that can be used to "close" completed, failed, or cancelled service requests as well as resolved incidents:

```
cd 'C:\Program Files\Microsoft System Center 2012\Service Manager\Powershell'  
Import-Module .\System.Center.Service.Manager.psd1  
  
$SomeDaysOld = (get-date).adddays(-7)  
$UNC_Now = (get-date).ToUniversalTime()  
$IncidentClass = Get-SCClass -Name System.workitem.incident  
$Inc_resolved = Get-SCClassInstance -Class $IncidentClass -Filter "Status -eq  
IncidentStatusEnum.Resolved" | ?{$_.ResolvedDate -lt $SomeDaysOld}  
  
If ($Inc_resolved -ne $null)  
{  
foreach ($Inc in $Inc_resolved)  
{  
$Inc.Status = "bd0ae7c4-3315-2eb3-7933-82dfc482dbaf"  
$Inc.ClosedDate = $UNC_Now  
Update-SCClassInstance -Instance $Inc  
}  
}  
  
$SRclass = Get-SCClass -Name System.WorkItem.ServiceRequest  
$SR_completed = Get-SCClassInstance -Class $SRclass -Filter "Status -eq  
ServiceRequestStatusEnum.Completed" | ?{$_.CompletedDate -lt $SomeDaysOld}  
$SR_failed = Get-SCClassInstance -Class $SRclass -Filter "Status -eq  
ServiceRequestStatusEnum.Failed" | ?{$_.CompletedDate -lt $SomeDaysOld}  
$SR_cancelled = Get-SCClassInstance -Class $SRclass -Filter "Status -eq  
ServiceRequestStatusEnum.Cancelled" | ?{$_.CompletedDate -lt $SomeDaysOld}  
  
If ($SR_completed -ne $null)  
{  
foreach ($SR in $SR_completed)  
{  
$SR.Status = "c7b65747-f99e-c108-1e17-3c1062138fc4"  
$SR.ClosedDate = $UNC_Now  
Update-SCClassInstance -Instance $SR  
}  
}  
If ($SR_failed -ne $null)  
{  
foreach ($SR in $SR_failed)  
{  
$SR.Status = "c7b65747-f99e-c108-1e17-3c1062138fc4"  
$SR.ClosedDate = $UNC_Now
```

```

Update-SCClassInstance -Instance $SR
}

}

If ($SR_cancelled -ne $null)
{
foreach ($SR in $SR_cancelled)
{
$SR.Status = "c7b65747-f99e-c108-1e17-3c1062138fc4"
$SR.ClosedDate = $UNC_Now
Update-SCClassInstance -Instance $SR
}
}
}

Remove-Module System.Center.Service.Manager

```

The above script would have to be run as a scheduled task either through the Service Manager workflow engine or as a server-based scheduled task. The same functionality can be provided through Orchestrator by using a runbook that leverages the Service Manager Integration Pack. The sample runbook shown in Figure 7-10 includes only resolved incidents and completed services requests; adding failed and cancelled service requests would require only a few more activities in order to be functionally equivalent to the Windows Powershell script above.

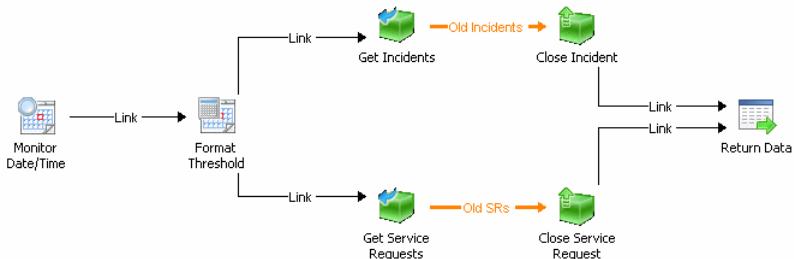


FIGURE 7-10 Runbook activity to close both completed service requests and resolved incidents.

The Format Threshold activity in this runbook provides the ability to "offset" from current time. This is important because whenever you are dealing with dates and times in Orchestrator and manipulating data in Service Manager, the date/time data is stored as UTC. You therefore should use the UTC option in Orchestrator. In Figure 7-11 the offset is "-5" which subtracts five days from the end of the previous activity time.

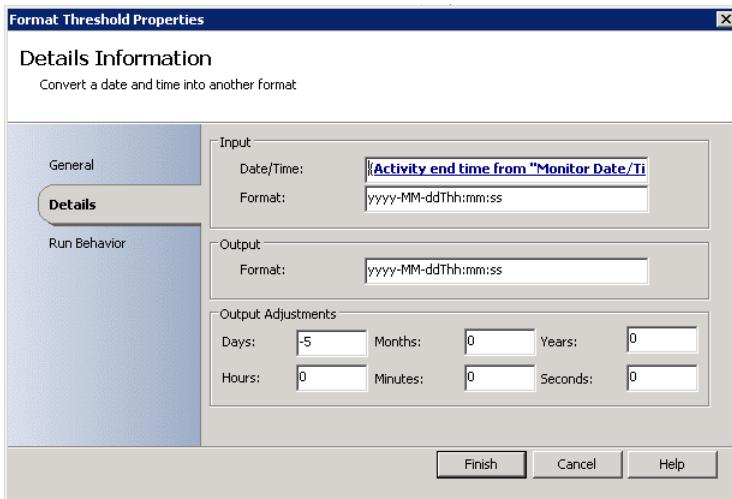


FIGURE 7-11 Runbook activity to subtract five days from current date/time.

The SCGetObject activity can be used to select service requests and incidents that meet the criteria of "Resolved" for incidents and "Completed" for service requests and respective dates greater than five days prior. Other criteria could also be added if necessary.

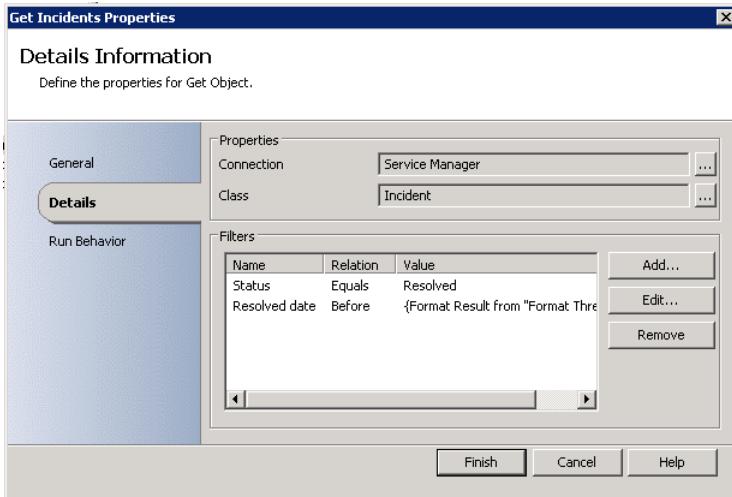


FIGURE 7-12 Runbook activity to get all resolved incidents that have been resolved five days prior than current date when running the runbook.

Finally, the UpdateObject activity can be used to update the work item. It's important to update Closed Date with the current UTC date and time since that is not set automatically.

This page left intentionally blank

About the authors

THOMAS ELLERMANN is a Principal Consultant at Microsoft Consulting Services (MCS) in Denmark. With more than 20 years of experience in IT, Thomas has worked for companies including Olivetti, HP, and Avanade. He has a broad knowledge within core infrastructure and has extensive experience in designing, migrating, and building medium and large enterprise environments.

In 2008, Thomas joined Microsoft where he has focused on systems management, working primarily with products like Microsoft System Center Service Manager, Operations Manager, and Orchestrator to deliver solutions to some of the largest customers in Denmark. Thomas also has a deep knowledge of Microsoft's cloud initiative and is a key contributor to developing Microsoft Datacenter Services solutions and deploying cloud projects worldwide.

Thomas co-authored a Microsoft System Center 2012 Service Manager training course, which he delivers across the world in conjunction with the Microsoft training organization. Thomas is also a regular speaker at various conferences.

Thomas writes regularly on his blog, which can be found on TechNet at <http://blogs.technet.com/b/thomase/>.

JOHN CLARK is an IT Service Management (ITSM) and System Center Architect with the Microsoft IT Service Management Practice.

John was formerly President of the Ohio Valley itSMF LIG, which achieved its third U.S. National itSMF IG Excellence Award in 2011. Both as an entrepreneur and an enterprise business IT professional, John has extensive background in IT management and IT service management, system and service automation, business process management (BPM), and enterprise architecture.

Since joining Microsoft, John has been recognized as a Microsoft System Center Service Manager resource. He leverages years of experience selling, designing, and implementing IT service management solutions that integrate across people, processes, and technology.

John has worked with and within IT organizations of varying sizes and has been involved with and focused on IT service management, enterprise architecture, and business process management for more than 25 years.

He earned an ITIL Service Manager Certificate in 2004, an ITIL v3 Foundations Certificate in 2008, and recently upon joining Microsoft, MOF 4.0 Foundations and ITIL v3 Expert certification.

He has published articles in industry journals and presented at various industry tradeshows on ITIL/ITSM, enterprise architecture, business process management and design, and BPM.

KATHLEEN WILSON is an Architect for the Worldwide Datacenter Center of Excellence (COE) team. Kathleen's focus is driving out solutions to support the Private Cloud, Service Provider, and the Enterprise initiatives. Prior to joining the Datacenter COE team, Kathleen was a consultant in Microsoft Consulting Services for seven years, focusing on Microsoft System Center Service Manager, Service Management, and Private Cloud.

Kathleen has been implementing service management software since 1998 when she received her first ITIL certification. She is now an ITIL v3 Expert. In the past, Kathleen was the President of the Toronto itSMF and served as the official reviewer of the ITIL v3 Service Transition book. She has been working with Microsoft System Center Service Manager since Beta in 2008. Kathleen has delivered Service Manager training to over 300 people and was a co-creator of the Service Manager 2010 and Service Manager 2012 Training Courses from Microsoft.

Kathleen has implemented Service Manager for many U.S. and Canadian customers. She currently is the internal WW Lead for Service Manager for Microsoft Services, responsible for internal readiness, sharing internal expertise and providing Service Manager guidance and coaching to the Microsoft Services team.

With over 15 years of implementing service management tools, Kathleen takes a practical, business-centric approach to implementing Service Manager for IT service management or Private Cloud use. Her focus on what works well out of box and on what Service Manager needs to achieve for the business results in success for many customers. Kathleen has also shared her knowledge of how to get the most from Service Manager implementations with a wider audience, presenting at the itSMF, Microsoft Management Summit, and to the U.S.-based Service Manager users group, SCSMUS.

KARSTEN NIELSEN is a Solution Architect, working the last 10 years for Microsoft Consulting Services in Denmark. Karsten has been in the IT industry for 17 years and in the consulting industry for 15 years. Karsten has focused on systems management for the last 12 years and has worked with Microsoft System Center Service Manager since the first version was in Beta. Over the years, the focus has shifted from technology to the processes and specialty business value. Currently the main focus is to work with companies to improve how they run their IT supported by System Center.

Karsten is a regular speaker on internal Microsoft conferences and has been part of developing several Microsoft offerings, such as the pre-assessment mentioned in this book.

About the series editor



MITCH TULLOCH is a well-known expert on Windows Server administration and virtualization. He has published hundreds of articles on a wide variety of technology sites and has written or contributed to over two dozen books, including *Windows 7 Resource Kit* (Microsoft Press, 2009), for which he was lead author; *Understanding Microsoft Virtualization Solutions: From the Desktop to the Datacenter* (Microsoft Press, 2010); and *Introducing Windows Server 2012* (Microsoft Press, 2012), a free ebook that has been downloaded almost three quarters of a million times.

Mitch has been repeatedly awarded Most Valuable Professional (MVP) status by Microsoft for his outstanding contributions to supporting the global IT community. He is a nine-time MVP in the technology area of Windows Server Software Packaging, Deployment & Servicing. You can find his MVP Profile page at <http://mvp.microsoft.com/en-us/mvp/Mitch%20Tulloch-21182>.

Mitch is also Senior Editor of WServerNews (<http://www.wservernews.com>), a weekly newsletter focused on system administration and security issues for the Windows Server platform. With more than 100,000 IT pro subscribers worldwide, WServerNews is the largest Windows Server–focused newsletter in the world.

Mitch runs an IT content development business based in Winnipeg, Canada, that produces white papers and other collateral for the business decision maker (BDM) and technical decision maker (TDM) audiences. His published content ranges from white papers about Microsoft cloud technologies to reviews of third-party products designed for the Windows Server platform. Before starting his own business in 1998, Mitch worked as a Microsoft Certified Trainer (MCT) for Productivity Point.

For more information about Mitch, visit his website at <http://www.mtit.com>.

You can also follow Mitch on Twitter at <http://twitter.com/mitchtulloch> or like him on Facebook at <http://www.facebook.com/mitchtulloch>.



Now that
you've
read the
book...

Tell us what you think!

Was it useful?

Did it teach you what you wanted to learn?

Was there room for improvement?

Let us know at <http://aka.ms/tellpress>

Your feedback goes directly to the staff at Microsoft Press,
and we read every one of your responses. Thanks in advance!



Microsoft