



This video is part of the

Microsoft Virtual Academy

For more information and to enroll for free
visit

www.microsoftvirtualacademy.com

Microsoft

© 2012 Microsoft Corporation. All rights reserved. Microsoft, Active Directory, Excel, Lync, Outlook, SharePoint, Silverlight, SQL Server, Windows, Windows Server, and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries.

The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.

Microsoft System Center
2012 - Service Manager
—Best Practices for
Data Center and
Private Cloud

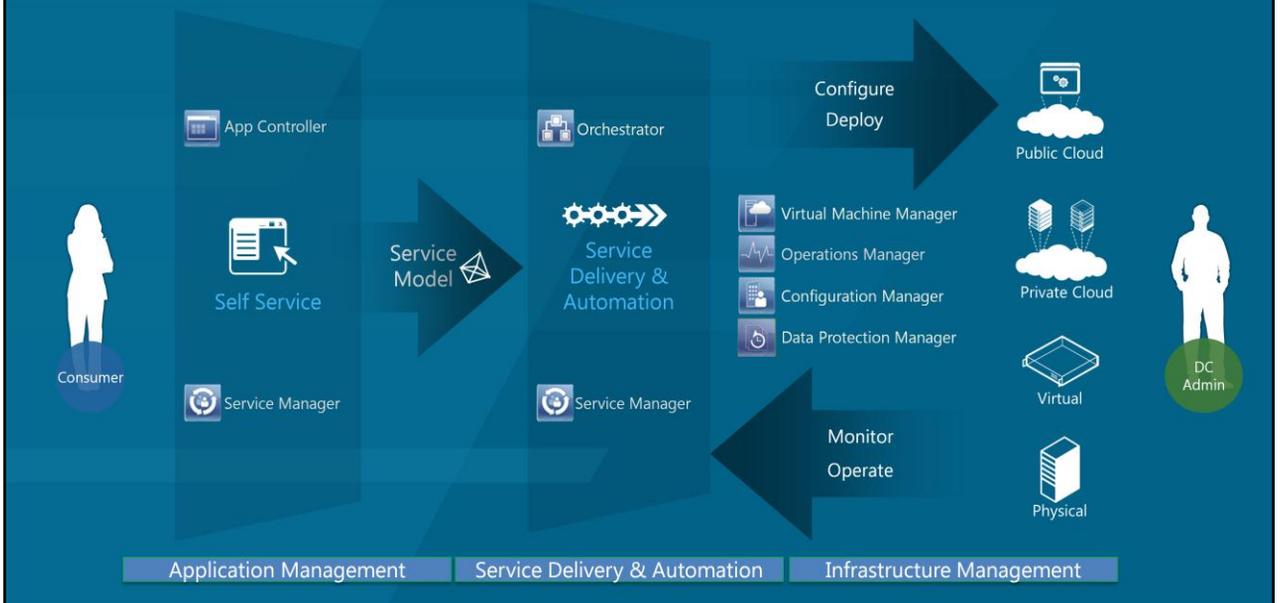
Travis Wright

Principal Program
Manager

Microsoft

This session discusses best practices for the private cloud and data center using Microsoft® System Center 2012 - Service Manager as well as other components of the System Center 2012 product, such as Operations Manager and Orchestrator.

System Center Helps Deliver IT as a Service

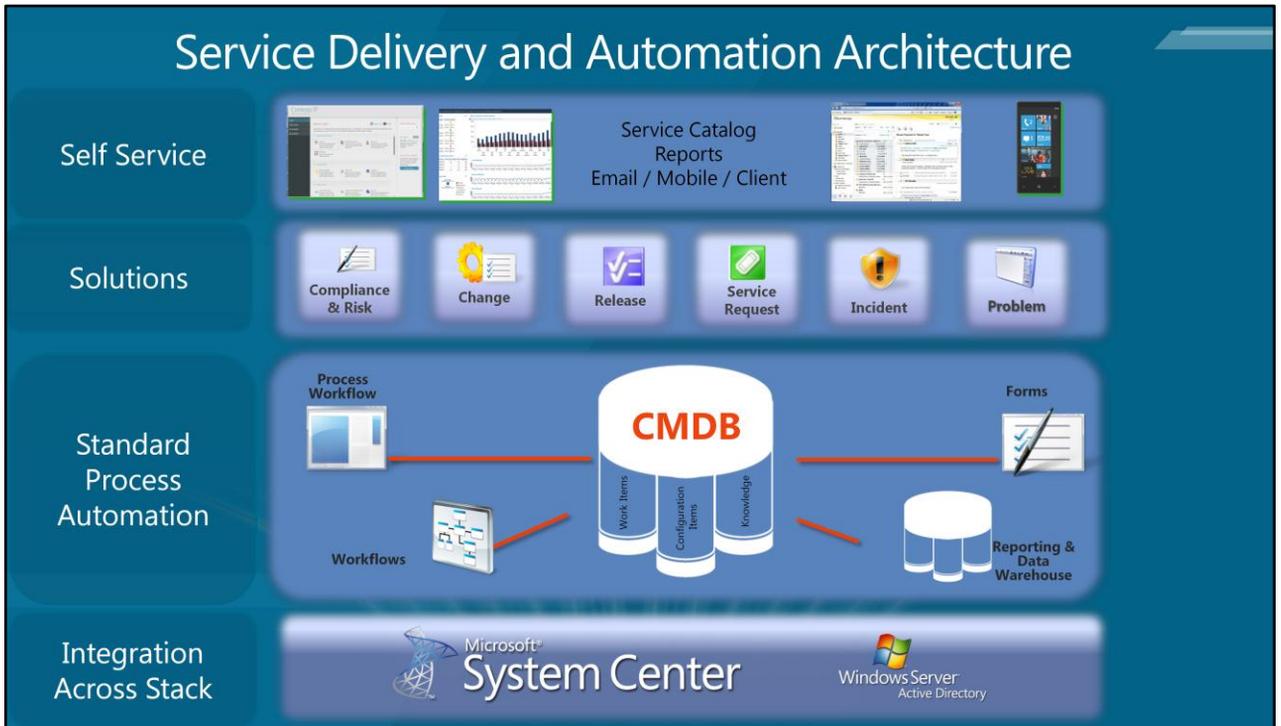


System Center 2012 delivers IT as a service through the data center administrator managing services delivered from the public cloud, private cloud infrastructure, and physical infrastructure. The components of System Center 2012 help to configure and deploy software onto the infrastructure, as well as monitor and operate things on a day-to-day basis.

Service Manager and Orchestrator sit at a layer up from the infrastructure and orchestrate some things across the other System Center components. Service Manager and Orchestrator provide service delivery and automation, and enable you to automate procedures in your IT organization.

System Center 2012 delivers services and ultimately provides a self-service interface to customers through its App Controller component and the Service Manager self-service portal.

Service Delivery and Automation Architecture



Service Manager:

- Has a CMDB database as part of its architecture to enable service delivery.
- Contains configuration items such as computers, the hardware around those computers, the software, and service components such as web sites and databases.
- Has a knowledge base that contains information in one CMDB.
- Allows you to create relationships between different objects, such as an incident and a database that went down.

Infrastructure:

- Workflow infrastructure can be used to optimize and automate the fulfillment of requests.
- The forms infrastructure can create customized forms, both on the self-service portal as well as in the Service Manager console.
- The data warehouse and reporting infrastructure can be used for long-term data storage and reporting purposes.

Connectors:

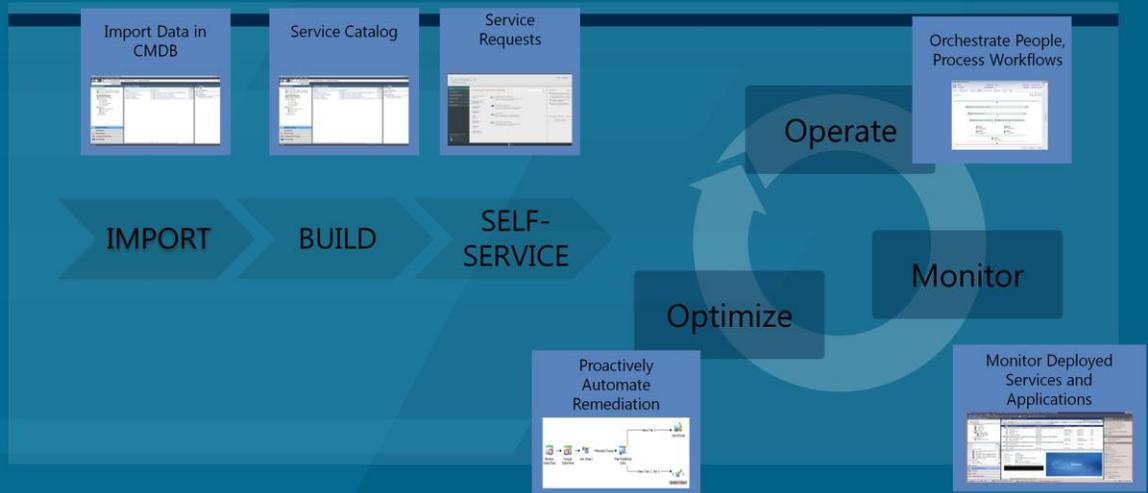
- Service Manager has connectors out of the box to other System Center components such as Operations Manager, Configuration Manager, Orchestrator, and Virtual Machine Manager.
- Service Manager also has out-of-the-box connectors to Microsoft Active Directory® directory services.
- These connectors bring data into Service Manager to drive the processes and enable Service Manager to automate activities across the System Center product.
- Service Manager provides a number of solutions out of the box for activities such as compliance and risk management, change management, release management, service request fulfillment, and incident and problem management.
- People can interact with Service Manager through the Exchange connector using either a mobile device or their Microsoft Outlook® or Outlook Web Access clients.

Self-service interfaces:

- Customers can now access the service catalog on the self-service portal, submit requests for incidents, and can view the status of those requests.
- Self-service reporting is also available, so people can go to either a Microsoft SharePoint® dashboard you create, or they can use tools they are familiar with such as Microsoft Excel® spreadsheet software, to analyze the data that is stored in the data warehouse.

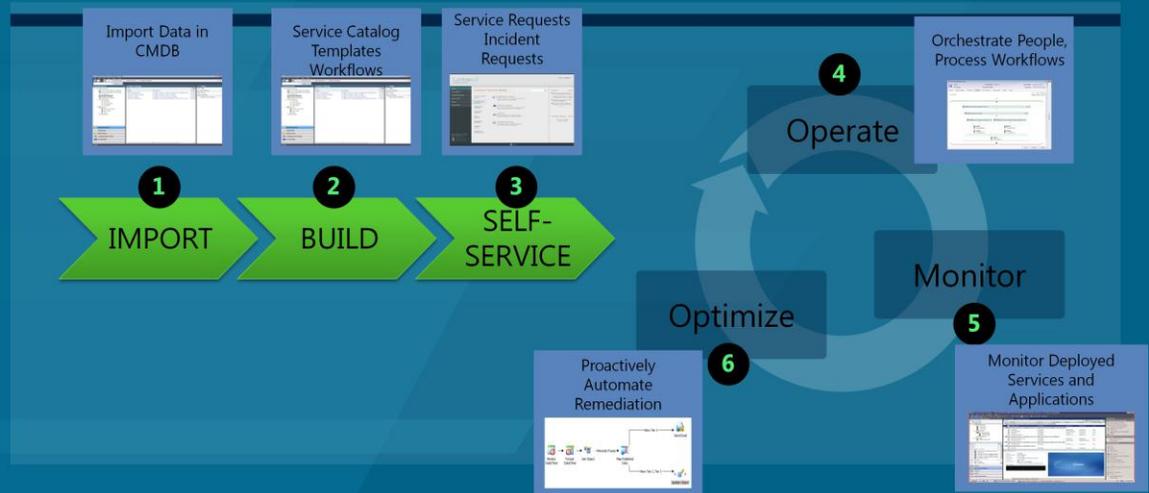
These solutions provide the forms, workflows, and reports that you need out of the box, and they are designed in such a way to be customized by you for your individual process.

Standard Practices



When using Service Manager, the first step is to import the data from various processes to the Service Manager database, and then build your requests that you plan to make available to people, and publish these to the self-service portal. Then you get into a continuous cycle of operating, monitoring for events, and optimizing your processes.

Standard Practices



The demo will show an example of importing data into the CMDB, and publishing back to the self-service portal as a service request. Then it will go through an operate cycle which orchestrates how the people in the process work together. Over time, it will monitor what is happening both in terms of workflows that are running and problems in the environment. Then based upon the experience, the processes will be optimized to automate and avoid doing something manually each time.

CMDB

Central to Service Delivery Architecture and Automation

Components

- CMDB
- Connectors

The CMDB is in Service Manager and can import various data from other sources. Out of the box, it has a connector to Virtual Machine Manager, which brings all the content you might see in the library of Virtual Machine Manager—such as service templates, virtual machine templates, storage, network, and load balancers—into the Service Manager CMDB. For example, when someone requests a virtual machine, instead of prompting them for how much RAM and CPUs they want, they can see a list of virtual machine templates from Virtual Machine Manager.

The Operations Manager connector brings in server and client information from those machines you are monitoring with Operations Manager to the Service Manager CMDB. The Service Manager CMDB and the Operations Manager database are built on the same technology infrastructure and have a shared data model. A management pack for Operations Manager that declares new classes and relationship types, for example for things that you are monitoring, can be imported into Service Manager with the exact same set of data as in the Service Manager database.

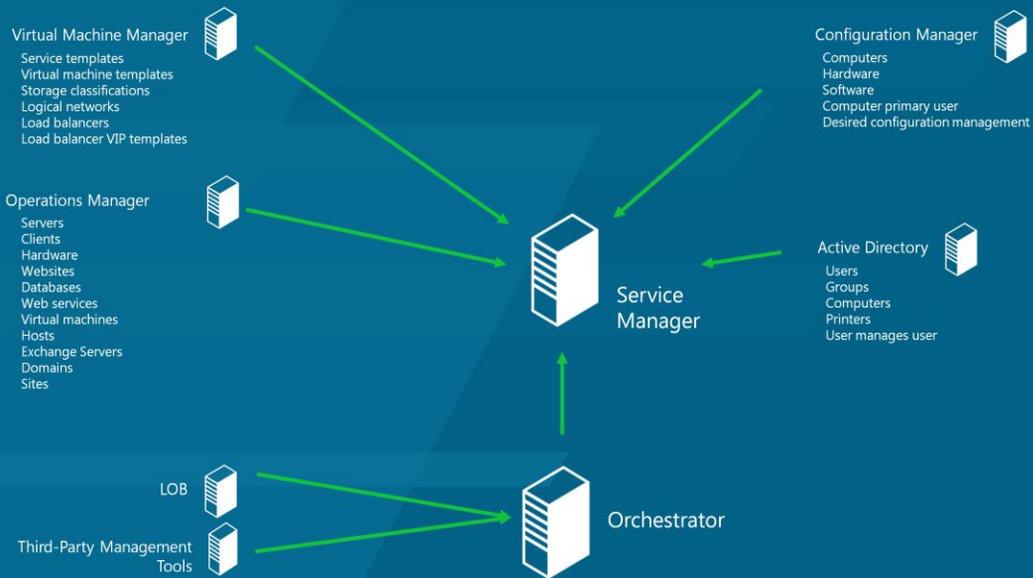
The Configuration Manager connector brings in data such as the computers that Configuration Manager is managing, and the hardware and software that are on them. Configuration Manager has the ability to look at logon history, and find who the primary user of a computer is, and bring that information to the Service Manager CMDB. With Configuration Manager, you can take anything that drifts out of a desired state configuration and create an incident in Service Manager automatically.

The Active Directory connector brings users, groups, computers, and printers into the CMDB.

The Orchestrator connector imports the runbooks into Service Manager. Any runbook that you have defined in Orchestrator can be brought into Service Manager, and you can use that runbook to automate a portion of a process to fulfill the service request, or remediate an incident for example. You can also use Orchestrator to create new runbooks or workflows which brings data from third-party sources into Service Manager. You can use any of the integration packs available for Orchestrator for other third-party management systems such as BMC, HP, or CA.

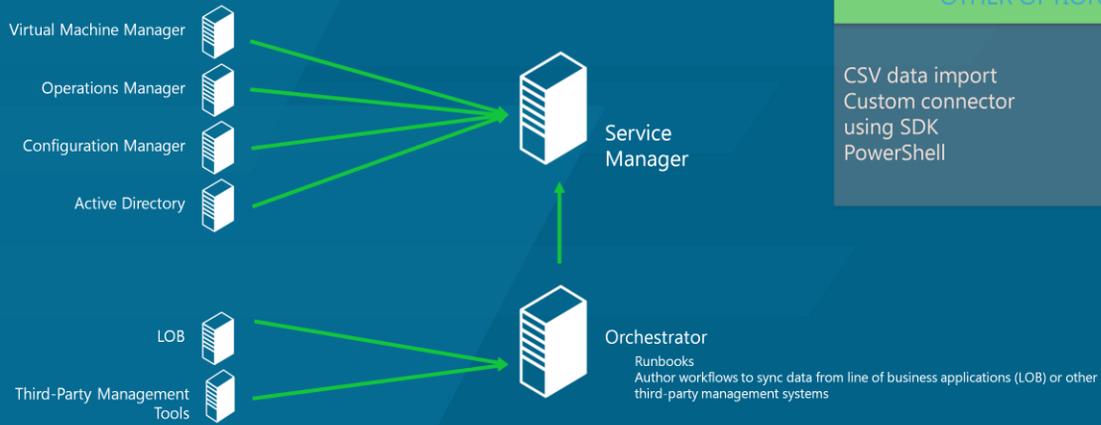
Generic activities such as a SQL query activity or web service activity can be used to go pull data and then use the Service Manager integration pack to insert that data into Service Manager. Other options for getting data into the Service Manager CMDB include CSV files for data imports. Custom connectors can be created with the Service Manager SDK. And PowerShell commandlets can be used to insert data.

Import



If you have data coming from multiple sources, sometimes you will have a situation where you have a single object that is being managed by multiple sources. For example, you may have a computer that's being managed by Configuration Manager and Operations Manager, and it's in Active Directory. This will work as you bring in all the relevant data from each of the different data sources, and map the data together onto a single object based on the key property value of that objects class.

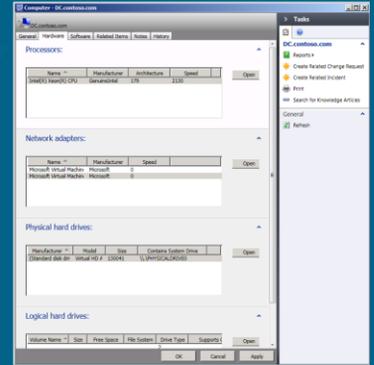
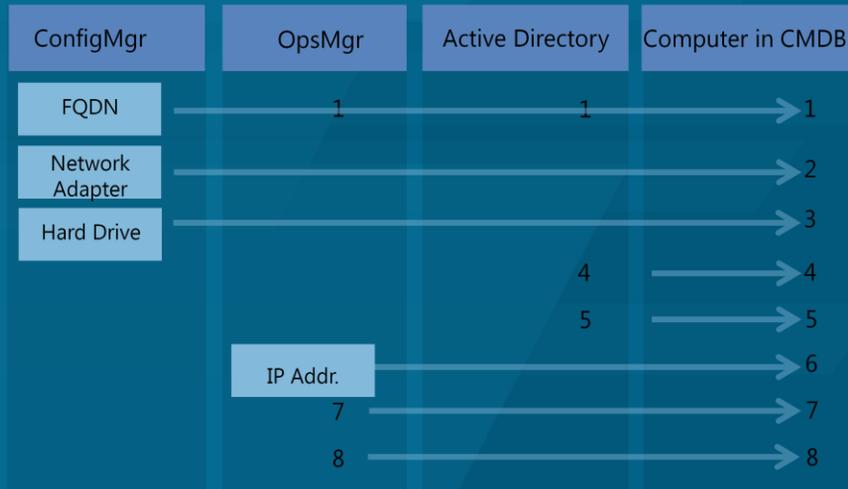
Import



For example, for the Windows® computer class, the fully qualified domain name is the key property for that class. Any time you bring in data for a computer, that computer can be found in Operations Manager and Configuration Manager, and it has the same fully qualified domain name.

In addition, you can merge the data together from those multiple data sources. Normally the data is the same coming from the different databases and data sources. In case there is a conflict, you resolve that through Service Manager, where the last writer wins.

Consolidated View of IT



This is an example of what the computer data will look like inside the Service Manager console.

SELF-SERVICE Service Catalog

Components

- ◆ Self-Service Portal
- ◆ Service Catalog
- ◆ Service Request
- ◆ Incident

The self-service portal is the framework for the user interface. The Service Catalog is where people can view the different service offerings and request offerings that are available for them to select. Requests end up as either their service request or an incident.

Self-Service Portal Features

- Service Catalog
 - Role-based access
 - Users fill forms to create service requests
 - Dynamic forms, flexibility
- Help Articles
 - Users can search and open knowledge
- My Requests
 - View/update own incident / service requests
- My Activities
 - View/update own manual activities
 - Review and approve



The service catalog is controlled using role-based security. You can set up user roles and grant different people different permissions on that service catalog. For example, you might have a request offering for onboarding a new employee; additionally, you may choose to make that available only to HR employees to request. You can use user roles to scope the visibility of these types of requests.

Once the user goes in, they can create a request by filling out a form. Forms can be dynamic; they're generated based upon the configuration that is specified so there's no need to write any code to create a form. There is no need to use a form designer; you just go through the wizard and specify the prompts for the user. You specify what type of data you expect to get back, and where you want to map those responses, in the service request or incident.

The self-service portal also features the ability for the end user to search the knowledge base and the help articles.

Users can create requests and see the status of that request in request view mode.

If a user is responsible for either manual activities or for approving requests, they can see those tasks on the My Activities page. Users can mark activities completed or vote on the review activity.

Technical Details for Service Manger Portal

- Web Content Server (WCS)
 - IIS sites and app pools
 - **Service host:** In-process client and server SDK, communicates by WCS to SMDB
 - **Content host:** Silverlight web parts packaged in XAP files
 - Navigation between web parts handled in config files
- SharePoint site
 - SharePoint Foundation 2010 or Enterprise
 - Can use existing SharePoint farms
 - Service Manager web parts come with .aspx wrappers that include customizations; enable SharePoint admin tools to customize
 - Add other web parts through SharePoint extensibility
- Supported configurations (installed through Service Manager setup):
 - SharePoint site and WCS co-located with Service Manager management server
 - SharePoint site and/or WCS remote from Service Manager management server

The Service Manager portal components include the web content server and a SharePoint site.

The web content server is an IIS site, and has an application pool. It has an SDK client, which is how it communicates between the server and the database. It also has the content host, which has a Microsoft Silverlight® web part that is delivered to the end user machine and run inside their browser. Where the user interacts with them, those calls are being sent back to the web content server host to get the data out of and post things back to the database.

The self-service portal in Service Manager is based on SharePoint 2010. This allows flexibility in customizing the look and feel of the portal. You can add additional web parts, pages, change the style, and perform other customizations.

You can install the portal website on an existing farm, or you can install this as another site on the farm . Alternatively, you can set up a new farm and run it that way. There are different supported configurations for running the website and web content server. For example, for development and test environments you could use the same machine as the Service Manager management server. However, it is recommended that you put the SharePoint site and web content server components on a separate server from the Service Manager management server as a best practice. You can also split up the web content server and the SharePoint site so these run on different machines as well.



Service Catalog and Self-Service Portal

demo

Follow along with this demo of the self-service portal and the service catalog.

In the Connectors view where you create and manage the different connectors, you can see the connectors for Virtual Machine Manager, Configuration Manager, and Active Directory. You have a connector for Operations Manager that brings in configuration item data, and another connector that brings in alerts into Service Manager as incidents. Then you have a connector to Orchestrator to bring in the runbooks.

In order to demonstrate how simple it is to create the connectors, here is a look at the properties on the Orchestrator runbook connector. Each of these connectors is very similar in terms of their configuration and provides a name and description, and optionally you can enable or disable it. With the Orchestrator runbook connector, there is a page to specify what you want to bring in from Service Manager. With Operations Manager, you choose which management packs to synchronize across classes. In the case of Configuration Manager, you choose which collections, and the scope of data to bring in.

It is simple to go through the wizard to create these connectors, and once you do the configuration, the data starts flowing into the Service Manager CMDB. There is no need to do any data mapping; this is all done for you as part of the connectors.

Looking at the configuration items workspace, you can see how some of this data has been brought in. As an example, you can look at the Computers view and see a list of computers that are brought in. If you look at the Operations Manager computer, you can see details such as the computers, identify, operating system, details on the status, and custodian. On the Hardware tab, you can see information on the hardware, such as disc drives, available space, and what software is installed. If you have Configuration Manager managing computers, you can see what updates are there, or missing. You can see data such as what virtual machine templates you have, and what virtual machines you have. Just bringing together all this data in one place is very powerful. From there, you can drive the automation and processes from one place.

As an example, here is a person that just joined a development team as a tester. His job is to test the "Dinner Now" line-of-business application that everybody in the company uses to order dinner when they are working late. From the URL, he can test the application running from the administrator page. When access is denied, he can make a request on the service catalog that will allow him to get permissions to the Dinner Now application group.

From the Service Manager self-service portal, he goes to the request offering. Normally when you get to the home page of the portal, you can look at a particular service offering, and see a list of request offerings that are available, and drill into one of those. In this example, because the user knows just what he needs to do, he goes directly to the page to request permission. This is a simple form for the user to provide their business justification. Once he submits, the form will create a service request into Service Manager. From this point on the requester can now view the status of the request. He can see the detail of the different activities in the process, and any notes in the action log.

Here is the perspective of the IT team. They now have a service request in their queue. When looking at open service requests, they can see that they have a new service request in progress. Here they can see the affected user, and the title that has been selected for them. This title was defined as part of the template that backs this request. Therefore, whenever somebody makes a request like this, it will always have this title set. This makes it easy to see what type of request it is. The urgency and priority are also set by the template. Again, there is consistency and standardization. It is always coming from the portal and account management area. User input is shown. They can see the business justification and the users' response.

On the Activities tab, they can see the review activity to approve the activity, and review activity to add the user to Active Directory. Opening up the review activity, they can see the users' business justification in the description field. This makes it easy for the approver to see the business justification and approve it.

Now the approver can approve it in the Service Manager console, or they can approve it in email using the Exchange connector, or through the self-service portal. They will be able to see the pending review activities, approve them, and save. Upon refresh, the review activity status changes to completed.

Next that triggers a workflow as an in-progress activity. The workflow triggers an Orchestrator runbook that will add the user to the group and Active Directory. Looking at the Orchestrator runbook, you can see the data activity, the username and activity ID, and how it passes the data into Service Manager. On the Runbook tab, the input parameters capture the login user name from the self-service portal, and the ID of the runbook activity workbook, and pass this information through. This information in Orchestrator will provide the group relationship. The runbook activity will see that there is a group related to the activity, as defined in the template. You create a relationship between runbook activity and the group object in the CMDB as part of the service request template. When in the runbook, you can get the group that is related to the work item, which shows the relationship instance, in the group object itself.

Using this group, you can now get it from Active Directory, look at the user from Active Directory and the username. Then you add the user to the group activity and add the user to the group. These two activities are Service Manager integration pack activities, and the other activities are coming in the Active Directory integration pack. This is a very powerful way to automate a process, just using simple drag and drop configuration experiences, without the need to write any code or scripts.

Opening up the runbook activity makes it easy to jump from the Service Manager console to the Orchestrator console to view the status of these jobs. In the Orchestrator console, you can see the runbook status. You can open up View Instance and see what is running. You can view the details and see the status as it runs through the process.

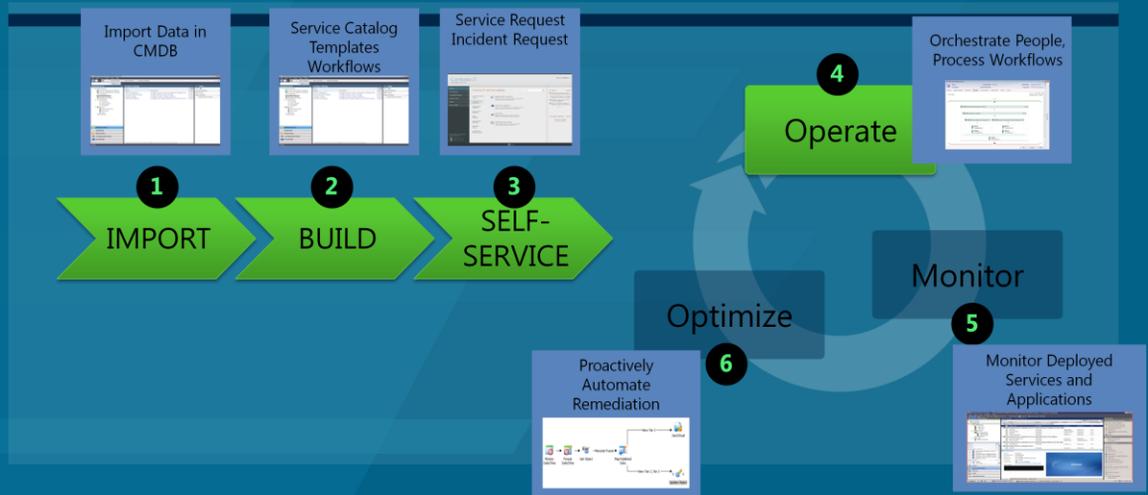
This is a good way to automate a simple task. It shows how automating simple procedures can reduce your costs and increase reliability. You can now see you have access to the application.

Now a PowerShell script will run to simulate a problem by disabling the website. Operations Manager will detect this, and create an incident in Service Manager. As a user, you will want to report this problem as well. At the self-service portal, in the service catalog, you can drill into the Dinner Now application, look at the service offering, and find the request offering that allows you to log an incident with the website. Looking at the landing page for the request, you fill the form, provide a description, and select the urgency.

Keep in mind these options are available to be configured by the IT administrator that is administering Service Manager. You can put the classification on the form, and make it read only. You want the users to know that you're going to classify this as an application problem, but you don't necessarily want them to change this. You can make these fields read only. You can also have the user attach a screenshot as well.

Once you create an incident, you can go to the Service Manager console, to the incident management workspace, and see the incidents generated from Operations Manager alerts. You can see the alert that the application is down, as well as the alert that the URLs are not being found, and you can see the incident that is submitted from the portal. You have all this information in one place.

Standard Practices



Standard practices diagram.

BUILD and OPERATE Service Level Agreements

Service Manager

- Calendars
- Service Level Objectives
- Service Level Agreements

When incidents are created, users expect you to respond in a timely manner. The best practice is to measure the service level agreements (SLAs) you have with your customers. A customer may expect you to respond to the incident and acknowledge you received it, and are moving to solve it. Depending on the type of request, you may have different SLAs.

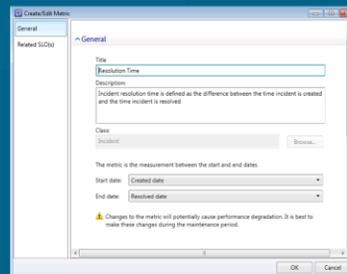
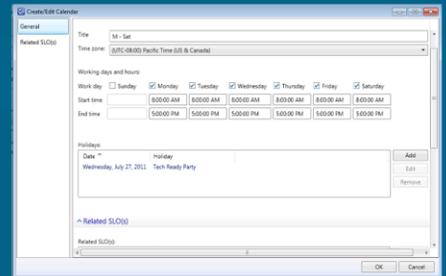
Service Level Agreements

Calendars

- Business hours
- Holidays
- Multiple calendars

Service Level Metrics

- In the box
 - Incident and service request
- Define your own time-based metrics



When you define a service level agreement in Service Manager, it uses the calendar to define business hours and holidays. Different groups can have different settings and options. The slide shows an example screen shot for configuring the calendar.

The other component of a service level agreement is the service level metrics, which is the difference of two time-based properties, for example, the difference between a created date and the resolved date. Microsoft provides several out-of-the-box metrics; you can add custom date and time properties and use those calculate service level metrics as well.

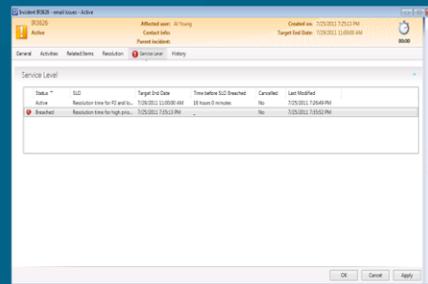
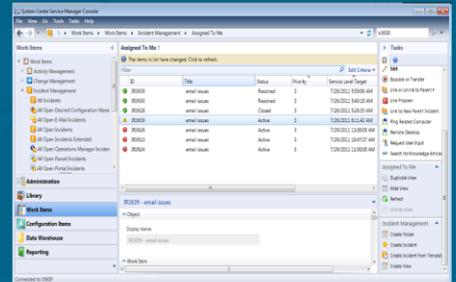
Service Level Agreements

Service Level Objectives (SLOs)

- Supported for all work items
- SLOs tied to pre-defined queues
- Specify target and warning thresholds

Notifications

- Views/forms showing color-coded breach/warning information
- Email notifications on warning and breach



A service level objective is something you define for what you are trying to track against. For example, for all the incidents in the queue, you can try to resolve these in a set time, and have a warning if you approach this time limit. You can be notified by email at the warning threshold or when it has been reached. System Center 2012 provides a number of views and forms out of the box that indicate the service levels you are working with, and what their status is.

Service Level Agreements

CALENDAR

- Business hours



SERVICE LEVEL METRIC

- Response time
- Resolution time



General

Related SLA(s)

General

Title: East Coast

Time zone: (UTC-05:00) Eastern Time (US & Canada)

Working days and hours:

Work day: Sunday Monday Tuesday Wednesday Thursday Friday Saturday

Start time: 9:00:00 AM 9:00:00 AM 9:00:00 AM 9:00:00 AM 9:00:00 AM

End time: 5:00:00 PM 5:00:00 PM 5:00:00 PM 5:00:00 PM 5:00:00 PM

Holidays:

Date: holiday

Buttons: Add, OK, Remove

Related SLA(s)

General

Related SLA(s)

General

Title: Resolution Time

Description:

Class: Incident

The metric is the measurement between the start and end dates

Start date: Created date

End date: Resolved date

Related SLA(s)

A service level agreement is a combination of a calendar plus a service level metric, plus service level objectives that determine the work item SLA that is being displayed to everyone.

Service Level Agreements

CALENDAR

- Business hours

+

SERVICE LEVEL METRIC

- Response time
- Resolution time

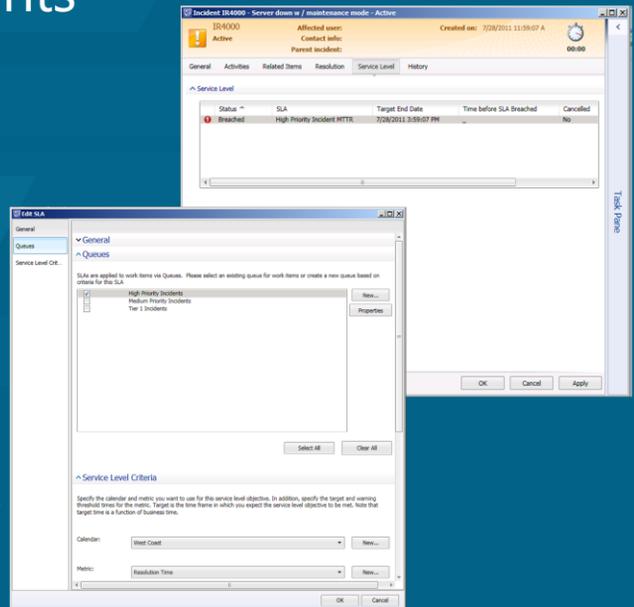
+

SERVICE LEVEL OBJECTIVE

- Work item (service request & incident by default)

WORK ITEM SLA

- Visibility
- Notifications
- Reactive processes



A service level agreement is a combination of a calendar plus a service level metric, plus service level objectives that determine the work item SLA that is being displayed to everyone.



Service Level Agreements

demo

Follow along with this demo of service level agreements in Service Manager.

When looking at the incident that was generated earlier in the self-service portal when the tester had a problem accessing the application, you can see on the service level tab there are two service level objectives that are being applied to this incident. One of them is to acknowledge the incident within a certain period of time. You can see that the service level agreement is about to be breached, and in addition, the amount of time you have to resolve the incident.

Looking at the service level objectives and how they are applied to the incident, you can see a defined calendar with the business hours set. The metrics have defined response time for a service request, a resolution time for incidents, and a completion time for service request and response time. Response time is the difference between the create date and resolved date.

The service level objectives can be applied to the enterprise applications queue. You can see the settings for the queue. You can see that the queue includes all incidents where the classification category equals enterprise application problems, and the priorities are greater than or equal to two. In this case, the incident falls into that queue. Therefore, it has the resolution and acknowledgement service level objectives applied to it.

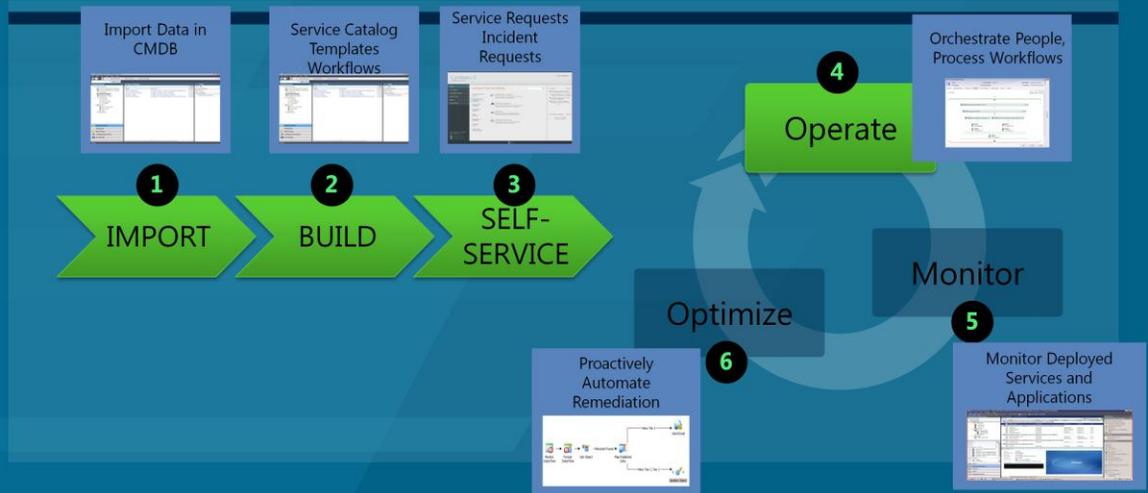
To prevent an alert on the incident, you can acknowledge the incident before time runs out on that service level objective. To resolve the incident, go to the console, look at a particular incident, and you can make it a parent request. Therefore, whenever the parent incident is resolved, the linked child requests will be resolved as well.

So now, you have a PowerShell instance that can restore the application. Going to the Dinner Now application, the service should work and the web page should display and work. As well, Operations Manager has detected that the application is up and has resolved the incidents that it generated automatically, and you can see that the child incident that was filed by the end user on the self-service portal has also been resolved.

You can see that the service level objectives for the target resolution time have been met as well. This shows how you restore service the way the user wants you to, and how you meet your service level objectives.

There were still some manual steps, which you can automate as well. To do that, you can create an Orchestrator runbook that will do the same thing that was done by manually running the PowerShell script. You would change the process to trigger this runbook when that incident happens. This happens through change and release management.

Standard Practices



Standard practices diagram.

OPERATE

Change and Release Management

Components

- Service Manager
 - Service Level Agreements
 - Change Management
 - Release Management

Change and release management has to deal with how to roll out changes to your system. A change request represents a single change to make to our system, and release management is used to control the process of rolling out multiple changes during a change release window.

You can roll up multiple change requests into one master quarterly, or monthly release. To visualize this, you can have multiple change requests linked to different projects and each of these change requests are mapped to a release.

Two new features in Service Manager include the ability to skip failed activities and skip any unnecessary activity in the process that you want.

There are new container work item types, such as the parallel work activity type that allows you to have a container into which you can drop other activities. There is a sequential activity, into which you can drop activities and those activities will happen in a sequence. By combining parallel and sequential activities, you can define any type of process with various logic in it.

There are two new configuration item classes, the environment and build class. Environment defines collections of configuration items to which the changes are being deployed, such as pre-production and test environments. The build class defines different builds of a line-of-business application that allows you to specify the build number and UNC file path for the build.

There are two new user role profiles, the release manager and activity designer; this allows you to provision what they need around release management inside Service Manager.

The new relationships are similar to what is in incident management, such that when you have a release record with child releases underneath it, once all the releases are complete, the parent will be completed. There is also a relationship that allows you to map a dependent activity, inside of a change request to an activity that is part of a release; this is how you create the relationship between the request and release record.

Change and Release Management in Operations

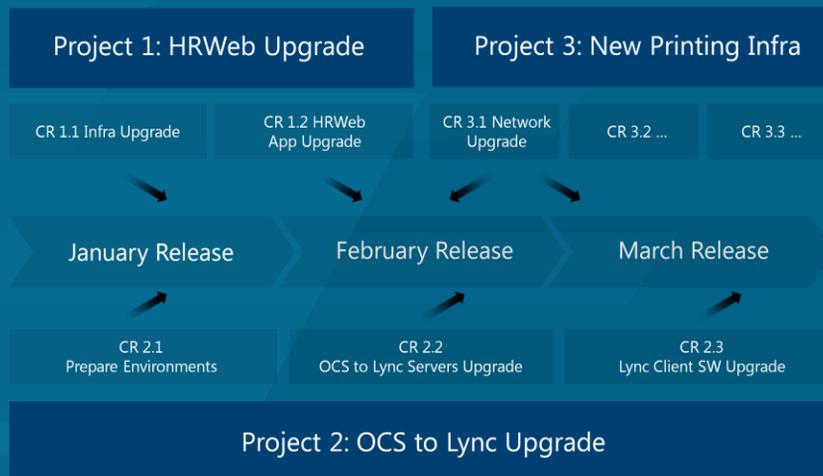
Change management makes decisions about individual changes

- Risk, impact, cost, feasibility, and so on

Release management deploys one or more approved changes to the controlled environments

Change and release management has to deal with how to roll out changes to your system. A change request represents a single change to make to our system, and release management is used to control the process of rolling out multiple changes during a change release window.

Change Management and Release Management



You can roll up multiple change requests into one master quarterly, or monthly release. To visualize this, you can have multiple change requests linked to different projects and each of these change requests are mapped to a release.

New Features

New workflow features

- Skip failed activity
- Skip unnecessary activity

Container activities

- Parallel
- Sequential

New CIs

- Environment
- Build

New Roles

- Release manager
- Activity designer

New Relationships

- Parent-child (similar to one in incident management)
- Change-to-release
 - Dependent

Two new features in Service Manager include the ability to skip failed activities and skip any unnecessary activity in the process that you want.

There are new container work item types, such as the parallel work activity type that allows you to have a container into which you can drop other activities. There is a sequential activity, into which you can drop activities and those activities will happen in a sequence. By combining parallel and sequential activities, you can define any type of process with various logic in it.

There are two new configuration item classes, the environment and build class. Environment defines collections of configuration items to which the changes are being deployed, such as pre-production and test environments. The build class defines different builds of a line-of-business application that allows you to specify the build number and UNC file path for the build.

There are two new user role profiles, the release manager and activity designer; this allows you to provision what they need around release management inside Service Manager.

The new relationships are similar to what is in incident management, such that when you have a release record with child releases underneath it, once all the releases are complete, the parent will be completed. There is also a relationship that allows you to map a dependent activity, inside of a change request to an activity that is part of a release; this is how you create the relationship between the request and release record.



Change and Release Management

demo

This demo shows how to create a change request and the process for automating the remediation procedures. It goes through the change request area of the console, and creates the change request with a template to standardize the process.

On the Activities tab, you can see the review activity and add in a dependent activity by using the activity form. This will show a placeholder that is referenced from the release record. Starting the process for the change request, you will see the review activity and the progress so you can review it. Switching over to release management shows that a release record has been created already. This is in the editing phase, and you can see the activities defined for this process and add in a new manual activity.

Once you have the manual activity defined, you can link to a change request activity and this will find all the change requests that are currently active and have dependent activities defined in them. You select the dependent activity to link to, and you can see the linked relationship by the icon. Once you kick off the activity, the review activity status will be in progress, and the notifications will be sent. Then once a manual activity is completed, it is linked to the dependent activity, and a workflow will automatically trigger in the background. Service Manager will make that dependency in the change request have its status changed to complete as well.

Next, the demo will show how to automate the remediation by kicking off an Orchestrator runbook that will do the same thing as the PowerShell script in the previous demo. From the Connectors view, you can change the alerts configuration to apply a different template, the incident alert auto remediation template.

Now when you simulate the error to cause the Dinner Now service to go down, the connector configuration in Service Manager will show that the templates have changed. You can see defined in the new template a new activity that is a runbook that will be called once it gets added to the incident, and that will call the Orchestrator runbook to pass in the incident ID and activity GUID, and then it will execute.

From the Orchestrator runbook, you can see the initialize data activity in it, similar to what you saw before, passing

in the data and activity information. Then it gets the activity using the activity GUID and the incident using the incident ID that is being passed in. Then it tests the Dinner Now application. If it comes up, then it will mark the incident resolved, and update the activity. If it can't resolve the incident, then it will update the activity as well. If the Dinner Now application is not working, it will reset IIS, and test the application. If it is still not working, it will reset the app pools, and test again. If it is resolved at this point, it will mark the incident resolved as seen earlier. If it is still not working, it will update the incident to add some information telling the administrator that it requires assistance.

An incident from Operations Manager shows this working. In this case, because the template has been changed, a new runbook activity is added to the incident. Therefore, this will trigger the Orchestrator runbook to automatically bring the website back up.

From the Service Manager console, you can see the incidents have been resolved. You can see the service level objective was met. So now you have automated the resolution of that incident by having a runbook triggered from the incident. Instead of applying a template that did not have the ability to call Orchestrator, you simply changed it to apply the right activity that had the template in it to automatically remediate this.

This is a way to reduce the amount of incidents in the first place. When there is an incident, you trigger the Orchestrator runbook. So chances are before anyone even noticed there was a problem, you have already resolved it so you do not have the additional incidents. Therefore, this is a great way to reduce your costs by automating these tasks.

This completes a review of all of the components of the Service Manager to manage your private cloud and your data center.



Microsoft®

© 2012 Microsoft Corporation. All rights reserved. Microsoft, Active Directory, Excel, Lync, Outlook, SharePoint, Silverlight, SQL Server, Windows, Windows Server, and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries. The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.