



Service Manager Integration Guide

version 2.4

© 2008-2011 Devfarm Software Inc.

© 2008 - 2011 Devfarm Software Inc. ALL RIGHTS RESERVED

This document contains proprietary information and other material protected by copyright of Devfarm Software or other 3rd parties. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use without written consent of Devfarm Software.

PowerWF Studio is furnished under a software license agreement and may only be used or copied in accordance with the terms of that agreement.

WARRANTY

Devfarm Software makes no claims as to the suitability of PowerWF Studio for any particular application. In fact, Devfarm Software specifically disclaims any implied warranty of the fitness of PowerWF Studio or any scripts or workflows created using PowerWF studio for a particular purpose. By its nature as a development tool, it is the responsibility of the workflow developer to continuously verify that scripts and workflows created with PowerWF studio are functioning as expected and to cease using said scripts and workflows if a problem is suspected. Devfarm Software shall not be liable for any direct, indirect, incidental, consequential, or other damage alleged in connection with PowerWF Studio or the related documentation

The information in this document and the software itself is subject to change without notice.

TRADEMARKS

PowerWF Studio and Visual PowerShell are trademarks of Devfarm Software Inc. Other trademarks and registered trademarks used in this document are property of their respective owners.

Devfarm Software
2008 Altom Ct.
Suite 100
St. Louis, MO 63146

Phone: 314.590.5800

Product Website: <http://www.powerwf.com>

Company Website: <http://www.Devfarm.com>

Blog: <http://blog.powerwf.com>

Twitter: <http://www.twitter.com/Devfarm>
<http://www.twitter.com/powerwf>

PowerWF Studio - Getting Started v2.4.012

Table of Contents

- Microsoft System Center Primer..... 4
 - Operations Manager..... 4
 - Service Manager 4
 - Configuration Manager..... 4
 - Data Protection Manager..... 4
 - Virtual Machine Manager 5
 - Opalis 5
- Why PowerWF Studio..... 5
- Getting Started With PowerWF Studio..... 5
- Requirements..... 6
 - PowerWF Running Remotely 6
 - Leveraging the Service Manager CMDLETS 6
- Tutorials..... 9
 - Tutorial: Integration With Service Manager Authoring..... 9
 - Topics Discussed 9
 - Procedure..... 9
 - Summary 20

Service Manager Integration Guide

Microsoft System Center Primer

Microsoft System Center is a family of products that help organizations manage, support and monitor their IT environments. System Center solutions can manage both physical and virtual environments across multiple datacenters covering a myriad of computers and devices.

The current System Center product suite includes:

Operations Manager

Operations Manager shows the state, health and performance information about devices and software systems in an IT environment. Operations Manager alerts on availability, performance, configuration or security situations.

Microsoft does a good job of providing coverage for Microsoft application with Operations Manager, but traditionally getting data out of third party or legacy systems required either purchasing management packs from ISV's or contracting developers to collect relevant data and write custom agents.

PowerWF makes monitoring these systems with Operations Manager almost trivial. The Operations Manager Integration Guide provides tutorials that discuss collecting data and creating management packs for custom and third party applications.

Service Manager

Service Manager is a relatively new addition to the System Center suite of products. Service Manager helps automate best practices within an IT organization, especially in cases of incident reporting and problem resolution, change control, and asset lifecycle management. Service Manager is tightly integrated with Operation Manager, Configuration Manager, and Active Directory Domain Services.

Service Manager provides a workflow based authoring tool to handle the logic behind the specific automated actions. PowerWF extends this functionality significantly by allowing users to either create custom toolbox items within the Service Manager authoring tool or to directly create Service Manager management packs. This guide includes tutorials that discuss the numerous integration points with Service Manager

Configuration Manager

Configuration Manager assesses, deploys, and updates software on servers , client computers and mobile devices. Currently there are minimal integration points between PowerWF and Configuration Manager, but as with every other Microsoft product, Configuration Manager is moving toward tighter integration with PowerShell. As this happens, PowerWF will integrate more closely with Configuration manager.

Data Protection Manager

Data Protection Manager provides unified data protection for Windows servers such as SQL, Exchange, and Sharepoint by providing disk, tape and cloud-based repositories for backup and recovery. Data

Service Manager Integration Guide

Protection Manager provides a number of PowerShell CMDLETS that can be used in PowerWF. Like every other PowerShell Snapin and Module, PowerWF will automatically discover these CMDLETS if they are installed on the PowerWF system and make them available as drag-and-drop activities.

Virtual Machine Manager

Virtual Machine Manager works as a hyper-visor for centralized management of physical and virtual machines within a data center. Virtual Machine Manager relies on PowerShell as their scripting management interface, so PowerWF can be used to easily monitor or manage what is happening within the Virtual Machine Manager's environment.

Opalis

Opalis is a recent acquisition by Microsoft to provide its users with an automation platform for datacenter operations. Opalis uses a workflow designer to create the automated tasks.

Although on the surface there appears to be similarities between Opalis and PowerWF, it quickly becomes apparent that they are complementary solutions, not competing solutions. One of PowerWF's strengths is the ability to deploy workflows as PowerShell cmdlets, Snapins, and Modules. Opalis allows users to extend Opalis using PowerShell, so it is pretty straight forward to create underlying, low-level automation with PowerWF and use Opalis for high-level orchestration.

Why PowerWF Studio

PowerWF Studio provides an easy way to create powerful automation scripts using a workflow paradigm and a simple drag and drop interface. Novice users can easily create valuable workflows by dragging the desired activities from the toolbox. Experienced users can create PowerShell scripts that are automatically converted into visual representations which can be further extended using the built in the activities.

Getting Started With PowerWF Studio

It is recommended that users walk through the tutorials in the PowerWF Studio™ Getting Started Guide before using this guide. The Getting Started Guide includes a number of tutorials that present basic concepts as well as more in-depth topics.

For more information about PowerWF's integration with System Center Operations Manager see the Operations Manager Integration Guide.

Service Manager Integration Guide

Requirements

In order to take advantage of PowerWF's integration with Service Manager, PowerWF should be run on a 64-bit machine running Windows Vista or later and using the 64-bit version of PowerWF. Be sure to install the Service Manager authoring tool on the PowerWF workstation. The authoring tool can be downloaded from Microsoft.com.

PowerWF Running Remotely

If PowerWF is not installed on the same machine as Service Manager, there are few required files that need to be copied to take full advantage of the integration between the two products. Administrator privileges will be required on both the PowerWF and Service Manager machines.

On the Service Manager machine locate the SDK binaries. These are typically located at:

C:\Program Files\Microsoft System Center\Service Manager 2010\SDK Binaries

Copy the entire directory to the PowerWF machine using this directory structure. This usually means adding directories for "Microsoft System Center" under "Program Files", followed by a sub-directory of "Service Manager 2010". The "SDK Binaries" directory lets PowerWF create workflows that are compatible with Service Manager.

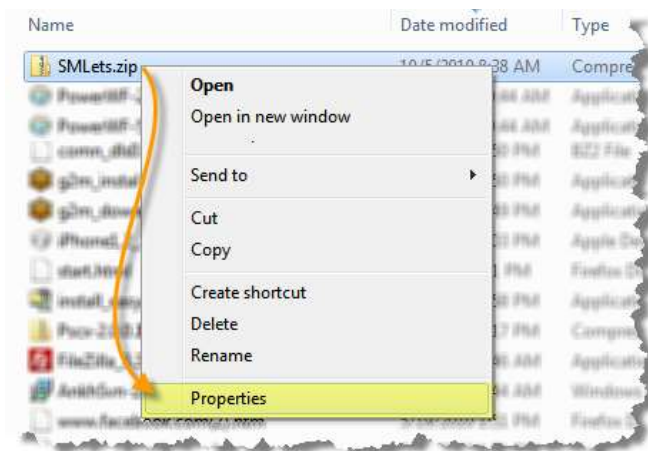
At this time, it is required that the SDK Binaries be installed on the C drive at:

C:\Program Files\Microsoft System Center\Service Manager 2010\SDK Binaries

Leveraging the Service Manager CMDLETS

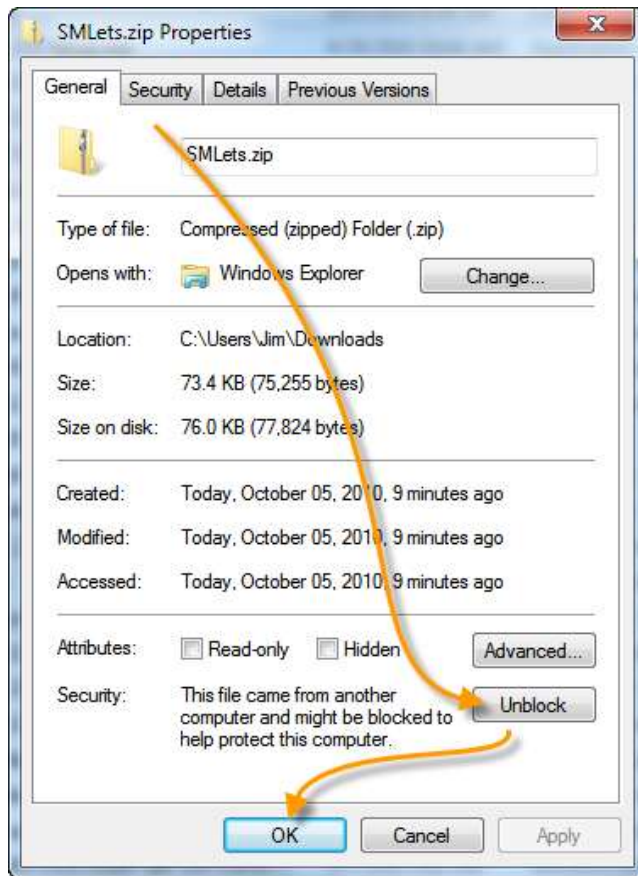
Regardless of where PowerWF is installed, it is recommended that the Service Manager CMDLETS be installed. Download the Service Manager PowerShell CMDLETS from <http://smlets.codeplex.com/>

Make sure the contents of the Zip file are unblocked. This is done by right clicking on the SMLets.zip file and selecting Properties.



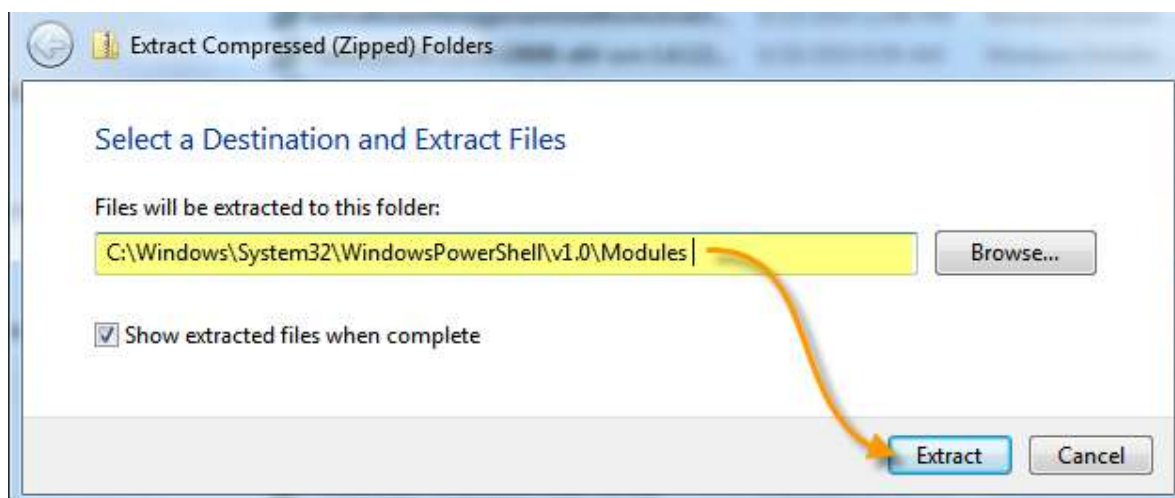
Service Manager Integration Guide

When the SMLets.zip Properties window appears, click the Unblock button and click OK.



Next unzip the CMDLETS in:

`C:\Windows\System32\WindowsPowerShell\v1.0\Modules`



Service Manager Integration Guide

To verify that everything is installed correctly, launch PowerShell with administrator rights. Run this command:

```
set-executionpolicy unrestricted
```

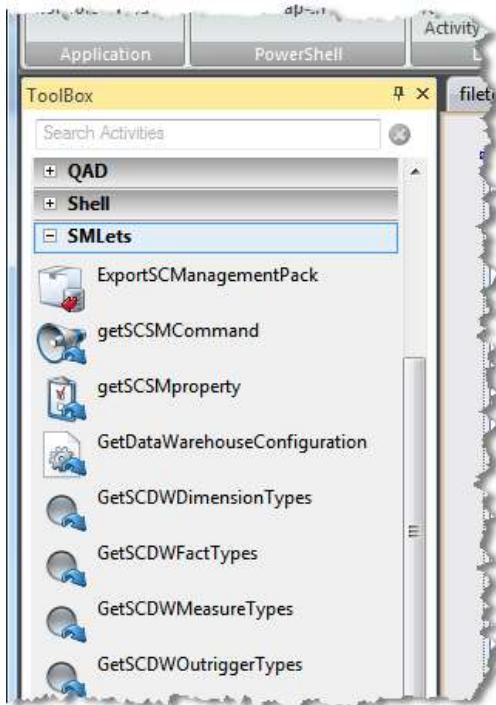
When prompted, click 'Y'.



The final step is to import the module in PowerShell:

```
Import-module smelts
```

As long the Import happens without errors, PowerWF should be able to discover the CMDLETS and use them as toolbox items.



NOTE: Be sure to run the 64-bit version of PowerWF when working with Service Manager

Service Manager Integration Guide

Tutorials

This section contains a number of tutorials that highlight PowerWF Studio's integration with Microsoft System Center Service Manager. Each tutorial includes a brief description of what the tutorial will accomplish as well as a list of topics discussed. The tutorial then provides step by step instructions with a brief explanation of why the step is performed. Each tutorial builds on the lessons of the earlier tutorials, including tutorials discussed in the Getting Started Guide, so tutorials in this Guide may glaze over steps such as creating a workflow or dragging and dropping an activity. In some cases, tutorials may be related to a specific technology, such as VMware. In these cases alternate tutorials are provided which cover similar topics.

Tutorial: Integration With Service Manager Authoring

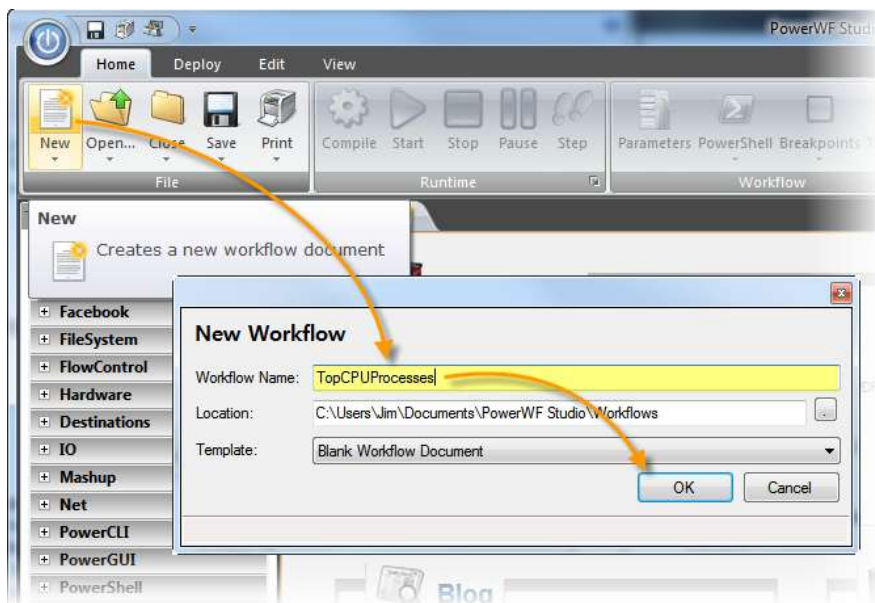
This tutorial discusses how PowerWF can be used to extend the existing Service Manager authoring tool by creating toolbox activities. The workflow will collect process information based on CPU usage.

Topics Discussed

- Creating a PowerWF Workflow
- Using the embedded PowerShell Editor
- Using Auto-Script Creation
- Creating Input Parameters
- Creating Service Manager Toolbox Activities
- Importing and Using Custom Toolbox Activities in Service Manager

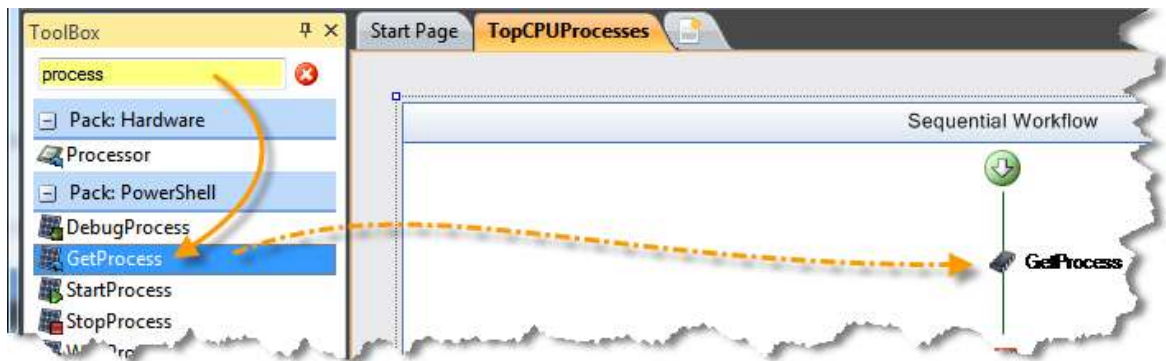
Procedure

1. Create a new workflow called "TopCPUProcesses"

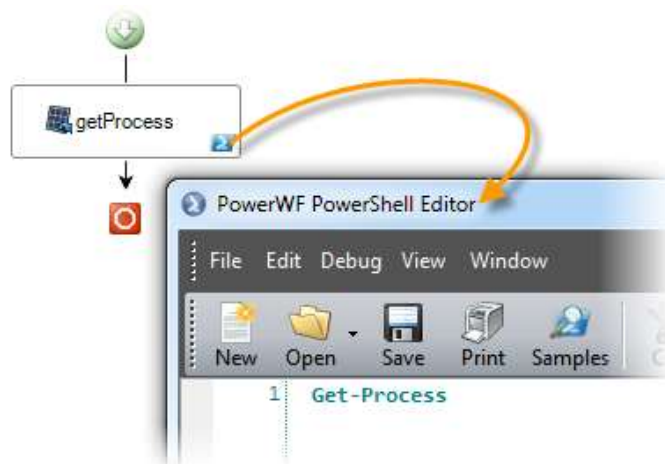


Service Manager Integration Guide

2. Type “Process” in the toolbox search window to find the Get-Process activity. As its name suggest, this activity collect information about the processes running on a computer.

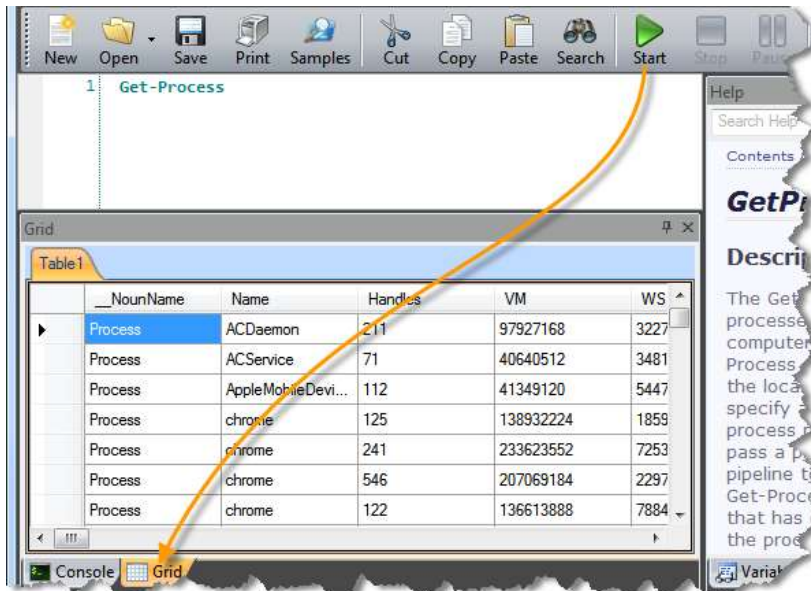


3. Double-Click the blue PowerShell icon on the getProcess activity to open the PowerShell editor.

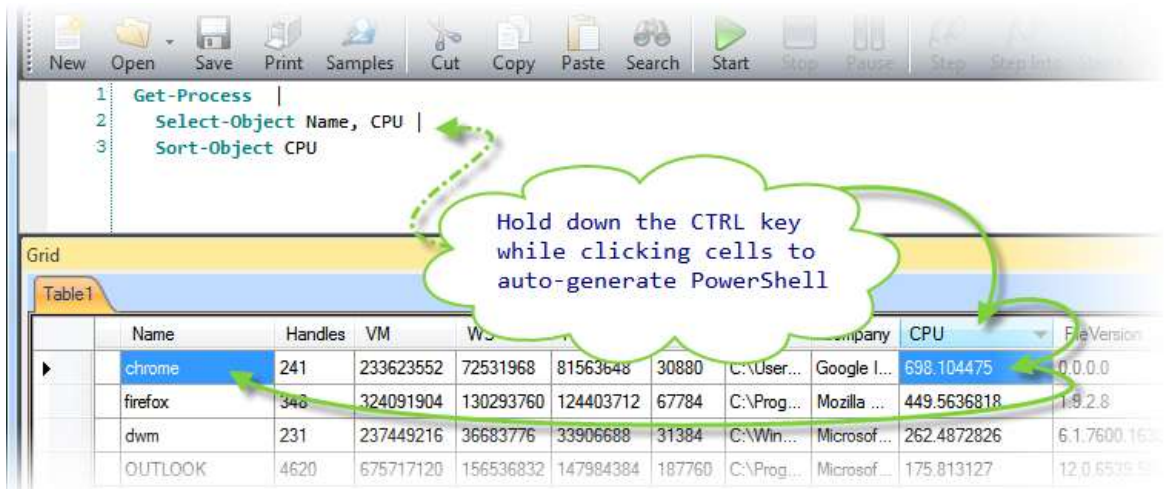


Service Manager Integration Guide

- Click the Start button to view the output of this activity. The Grid tab shows the complete results in a table format.



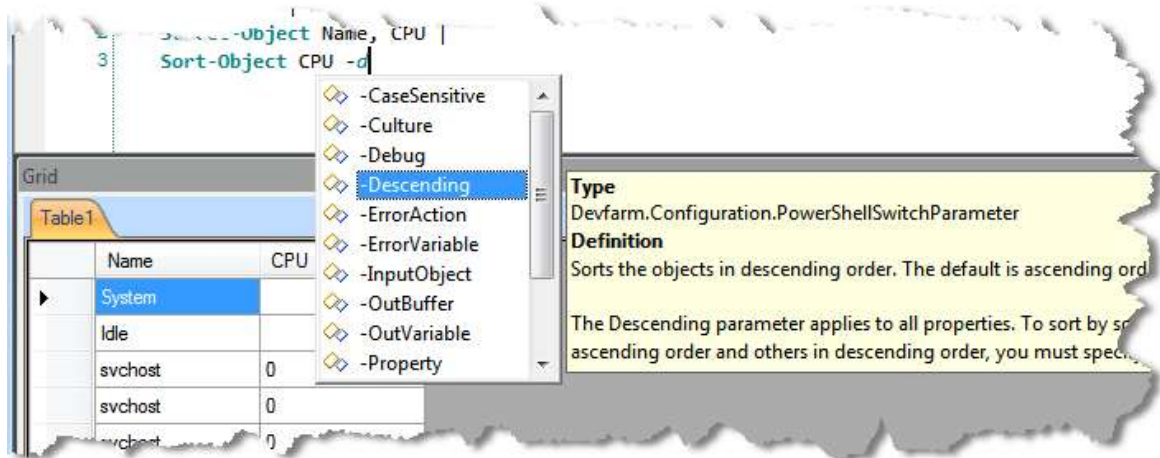
- One nice feature of PowerWF is the ability to automatically generate PowerShell script based on user interaction with the Grid pane. Find the CPU column and click the heading to sort the results. Next, while holding down the CTRL key, select a cell in the CPU and Name columns.



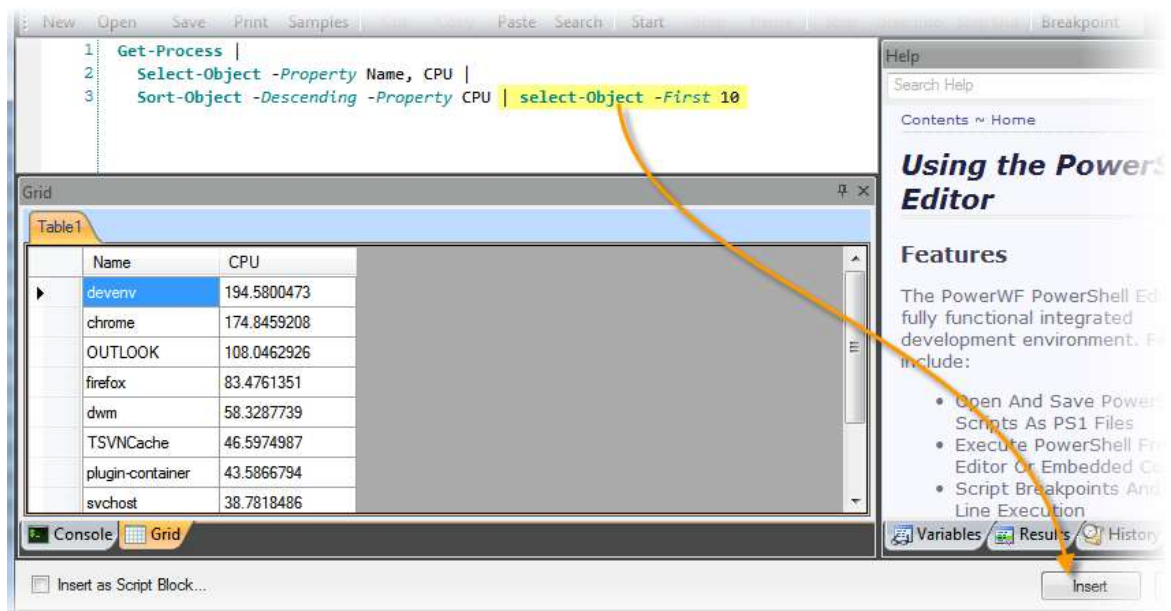
Note: If desired, additional cells can be selected. Be sure to hold down the CTRL key the entire time while selecting columns.

Service Manager Integration Guide

- Click the Start button to replay the PowerShell script. One thing that will be obvious is that the CPU should be sorted in descending order. Type `-d` after the word CPU and select `-Descending` to sort the results based on High CPU usage.

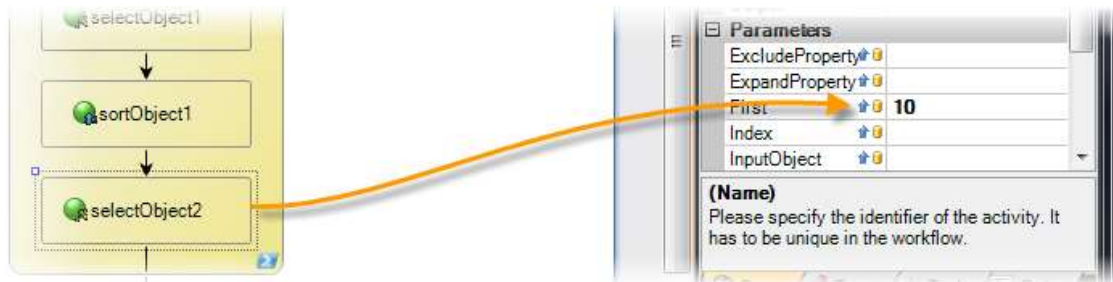


- To limit the results to the top ten CPU consumers, add a `| select-Object -First 10` to the end of the script then click the Insert button to add this PowerShell script to the workflow.

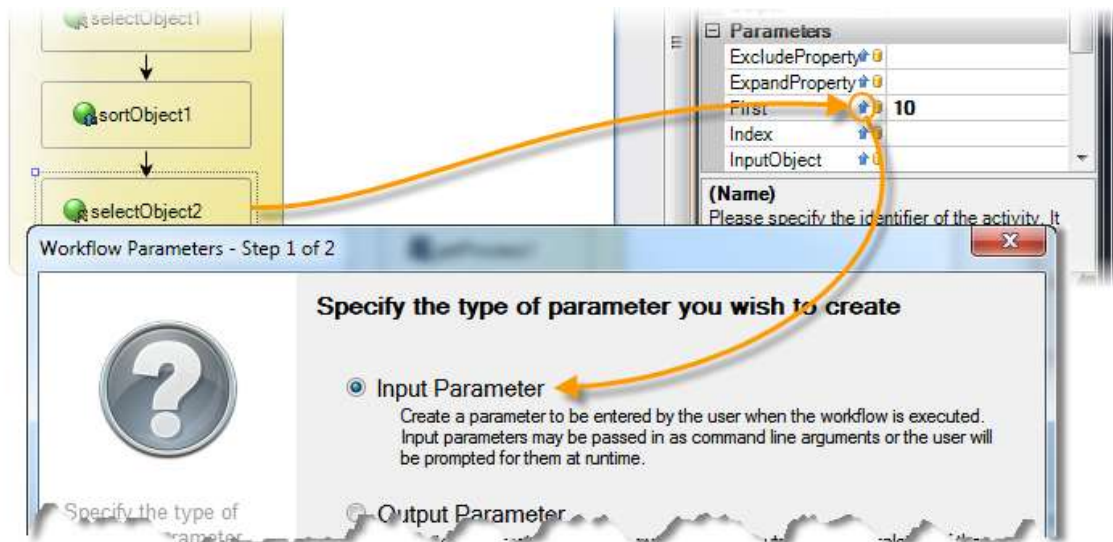


Service Manager Integration Guide

- PowerWF takes the PowerShell script and converts it into Workflow components. The workflow consists of a single pipeline with four PowerShell CMDLETS. Click on the “SelectObject2” activity. The Property view shows the properties for the selected CMDLET, including that the First 10 items will be selected.



- Double-click the blue arrow icon to create an Input Parameter for the number of processes to select. After selecting Input Parameter click Next.



Service Manager Integration Guide

10. Set the Input Parameter name to “NumberOfProcesses” and the Help Text to “Specifies the number of processes to display”. Also mark this as a Mandatory Variable and hit Finish.

Workflow Parameters - Step 2 of 2

Input Parameter

Name: NumberOfProcesses

Help Text: Specifies the number of processes to display

Initial Value: 10

Data Type: System.Object

Scope: Input

Mandatory Variable Accept Parameters

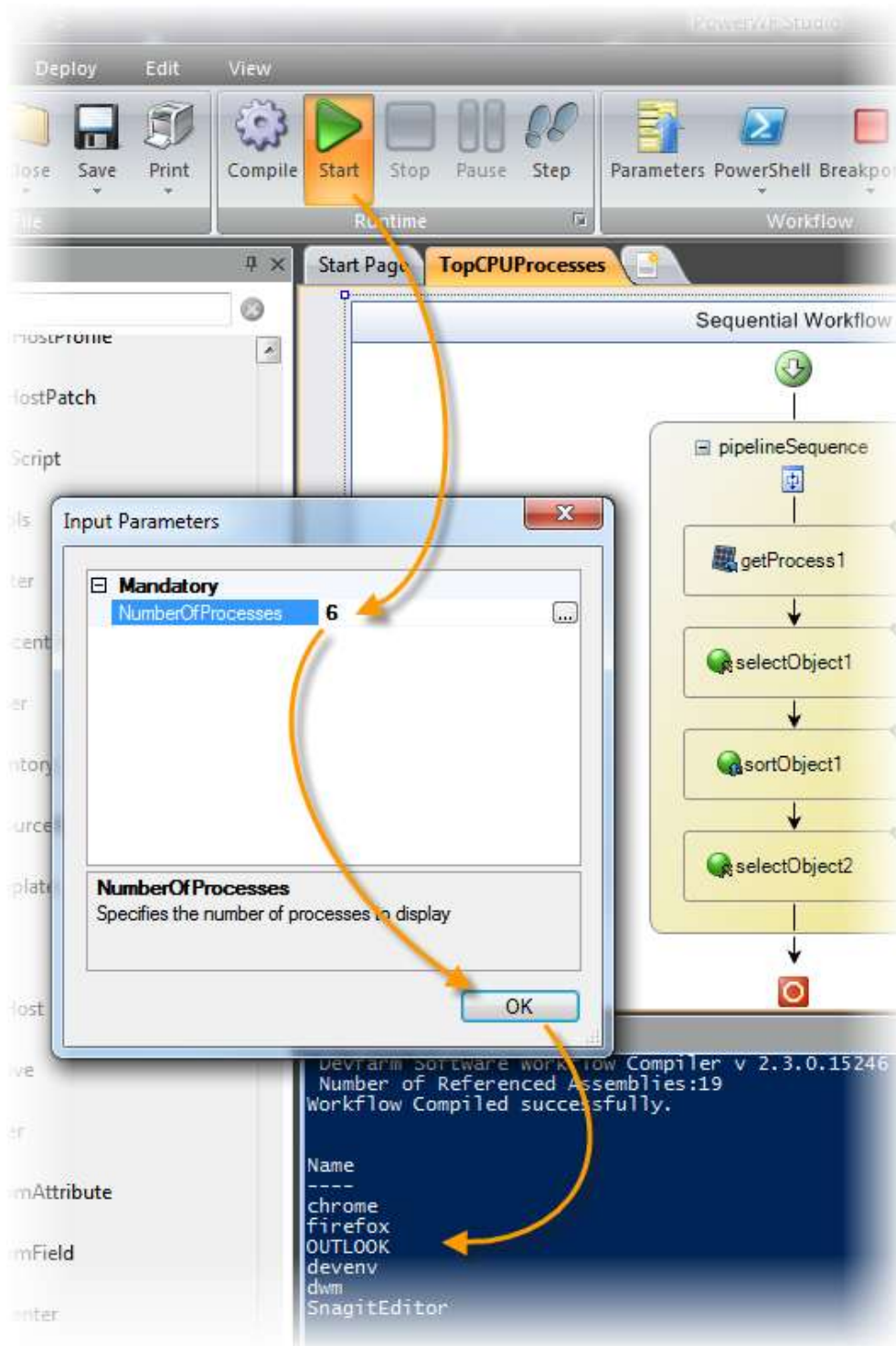
Enter a meaningful name and specify if the parameter will be used as an input or output. Use Help Text to explain the purpose of this parameter.

NOTE: Input parameters can be

NOTE: The workflow will now have an input parameter that must be entered before the workflow will run. When running the workflow from PowerWF Studio, the user will be prompted to enter the desired number of processes to return. Once this workflow is deployed as a CMDLET, the Input Parameter can be entered as a command line argument. In this tutorial, the workflow will be deployed to Service Manager where the Input Parameter will become a property that the user can enter.

Service Manager Integration Guide

11. Run the workflow again, specifying 6 for the number of processes to return. The output will appear in the Console output.

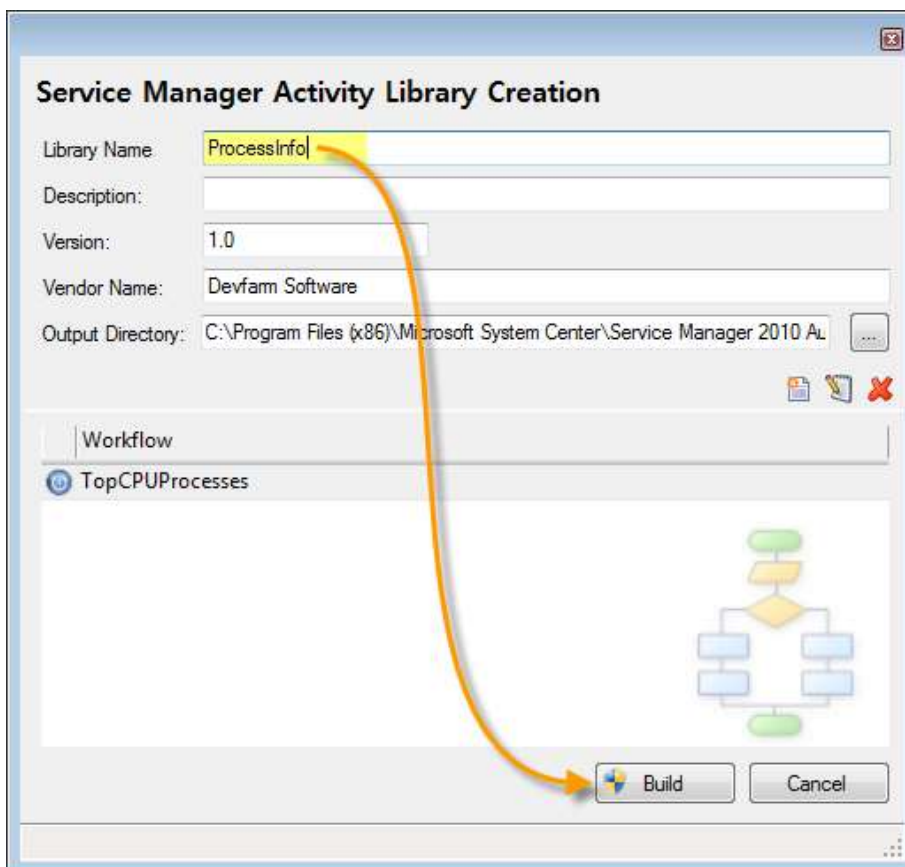


Service Manager Integration Guide

- Now that the workflow is complete, it can be deployed as a Service Manager Activity Library. On the Deploy tab, click the Service Manager button and select "Create Activity Library".

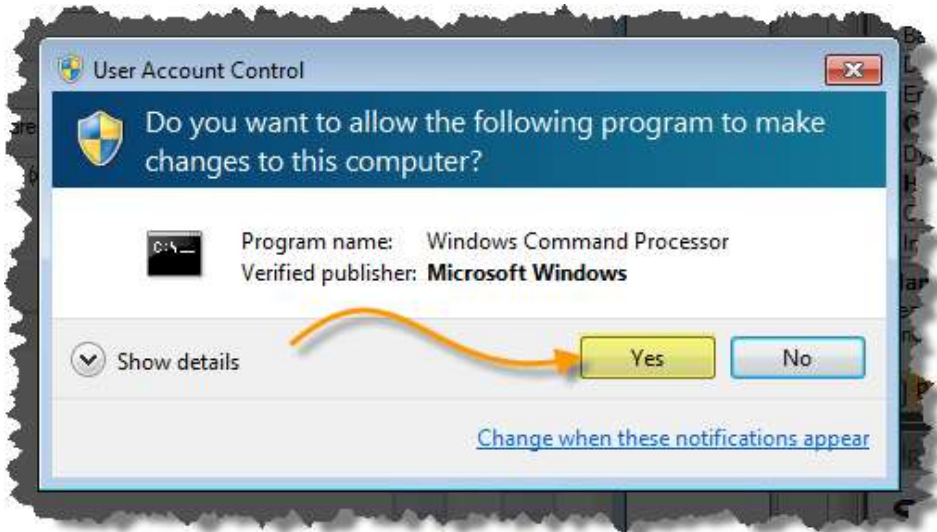


- The Service Manager Activity Library Creation dialog lets you specify a Library name and which workflows to include in the activity library. Set the Library Name to "ProcessInfo" and click Build.

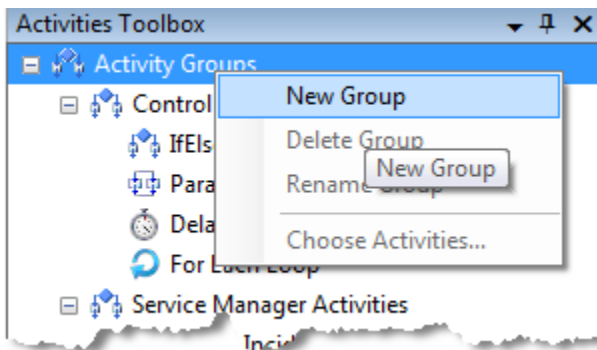


Service Manager Integration Guide

14. PowerWF will now build the Activity Library. If prompted, click “Yes” to allow PowerWF and Windows Command Processor to make changes to the computer.



15. Launch the Service Manager Authoring Tool. In the Activities Toolbox, Right-Click on Activity Groups and select New Group.



16. Name the new group, “PowerWF”

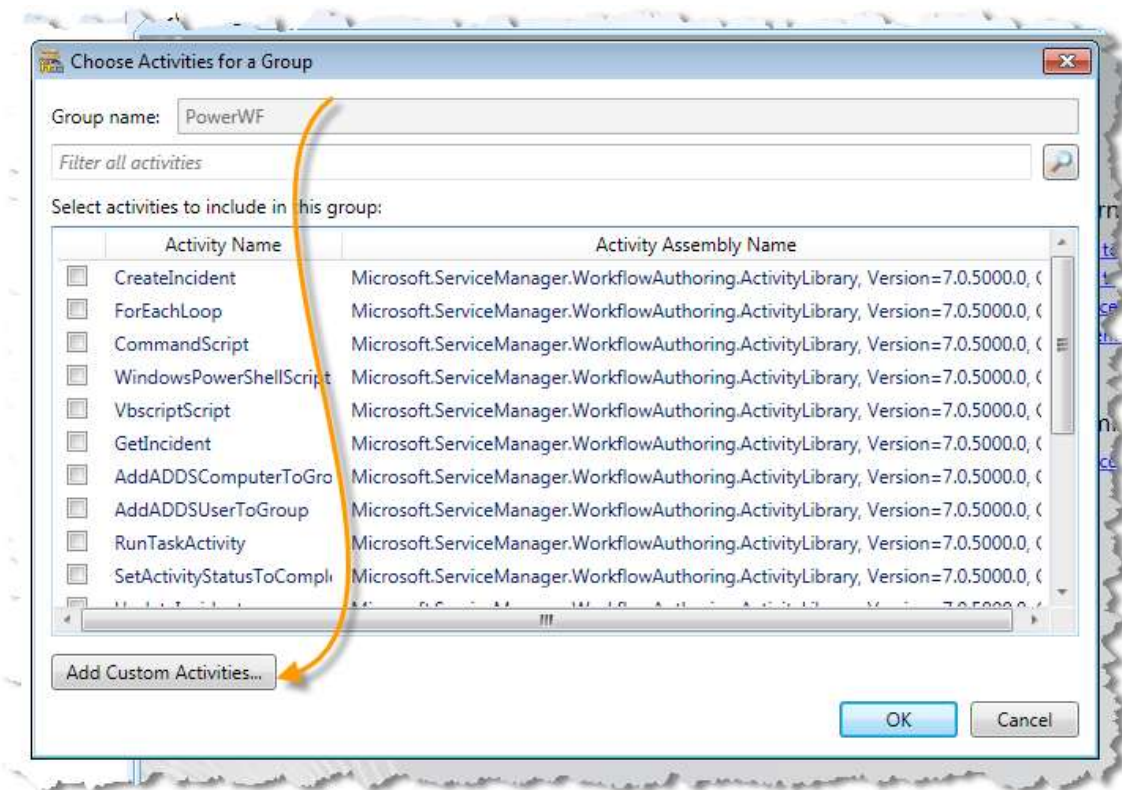


Service Manager Integration Guide

17. Right click on the new “PowerWF” group and select, “Choose Activities...” from the pop-up menu.

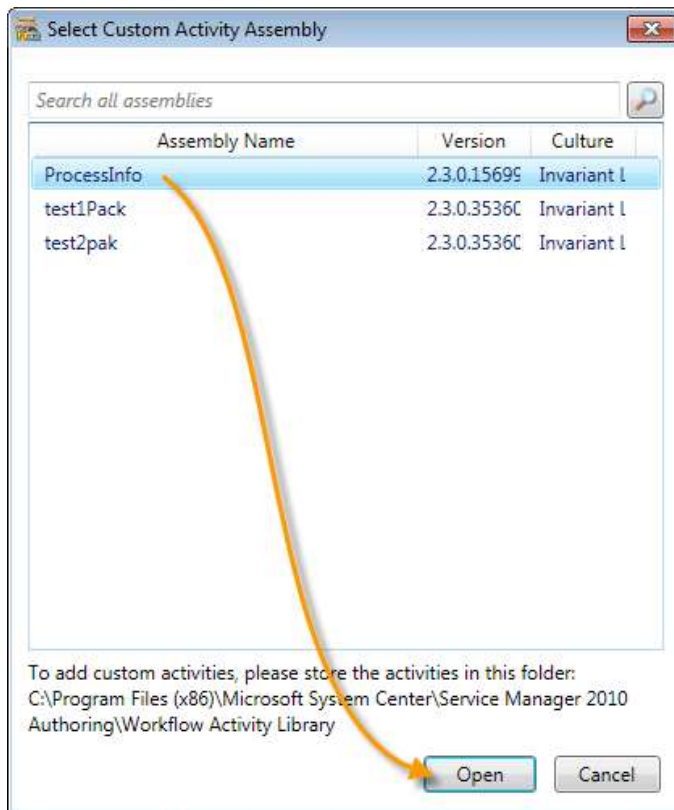


18. When the “Choose Activities for a Group” dialog appears, click the “Add Custom Activities...” button.

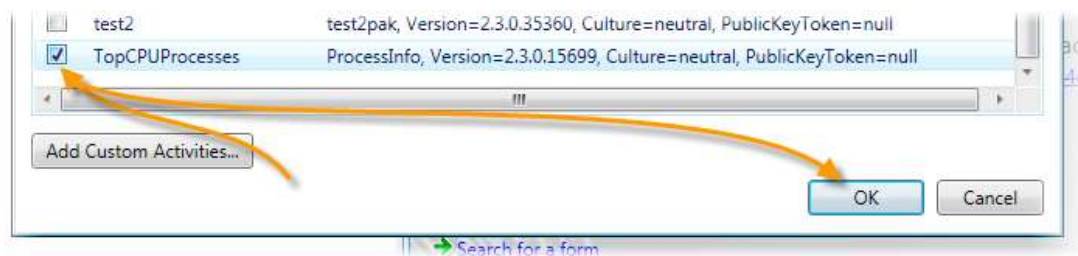


Service Manager Integration Guide

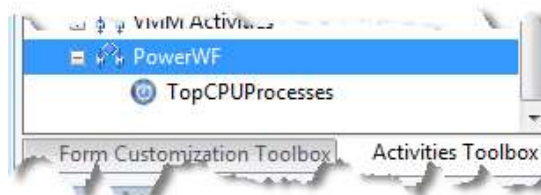
19. Select “ProcessInfo” from the “Select Custom Activity Assembly” dialog and click Open. Then



20. When the “Select Custom Activity Assembly” dialog closes, be sure to select the “TopCPUProcess” activity from the “Choose Activities for a Group” dialog and click OK.

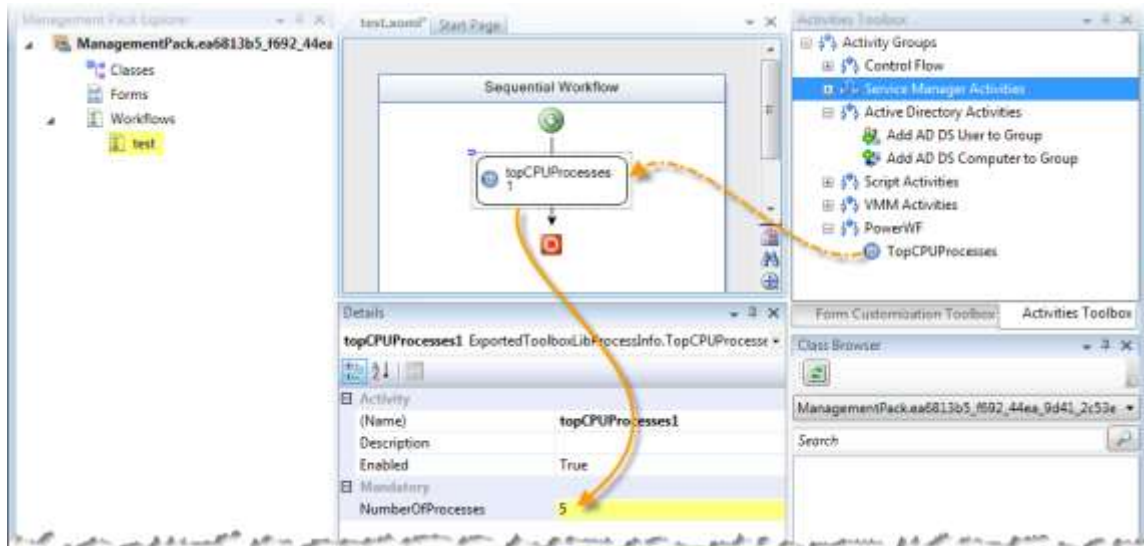


21. The workflow created in PowerWF is now available for use in the Service Manager Authoring Tool.



Service Manager Integration Guide

- To use this new activity, open or create a management pack in the authoring tool and select a workflow. Drop the TopCPUProcesses on the workflow and set the “NumberOfProcesses” to retrieve.



Summary

This tutorial showed how to create a Service Manager Activity Library from a simple PowerWF workflow. This tutorial touched on some PowerWF basic concepts, such as creating a workflow, using the PowerShell editor and automatic PowerShell script creation. An Input Parameter was created and the workflow tested. The workflow was deployed as a Service Manager Activity Library and then imported into the Service Manager Authoring tool. The tutorial showed a simple case of using the imported activity.

Service Manager Integration Guide

Tutorial: Creating a Scheduled Workflow

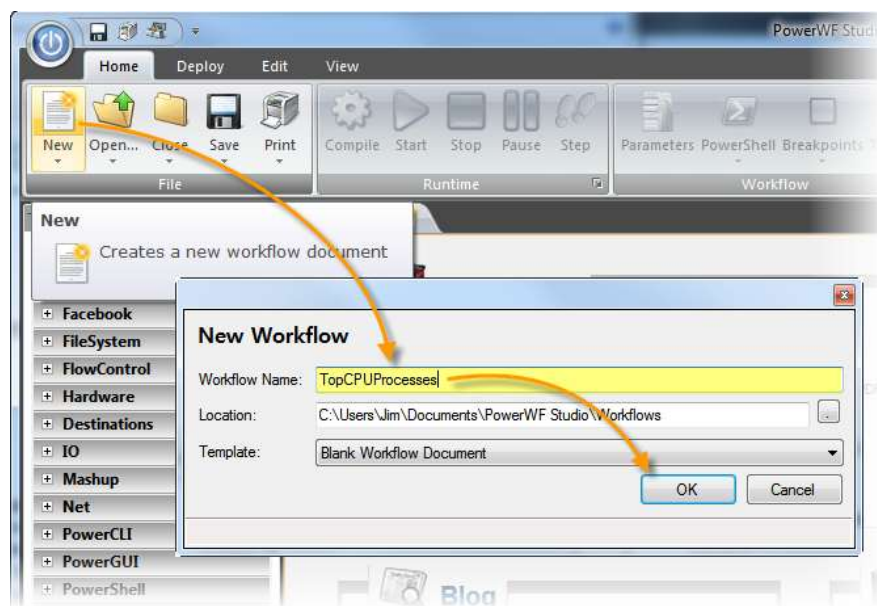
This tutorial discusses how to create a scheduled workflow in Service Manager using a PowerWF workflow. The workflow created for this tutorial will create a simple report and email it to a user.

Topics Discussed

- Creating a PowerWF Workflow for Service Manager

Procedure

1. To create a workflow for Service Manager



Summary

This ...