CA Release Automation VMware vSphere Actions Shared Components Release Notes

Author:	Michael Long
Version:	0.2
Filename:	CA VMware Shared Componentsv0.2.docx
Date:	March 1, 2015

Table of Contents

What is a Shared Component

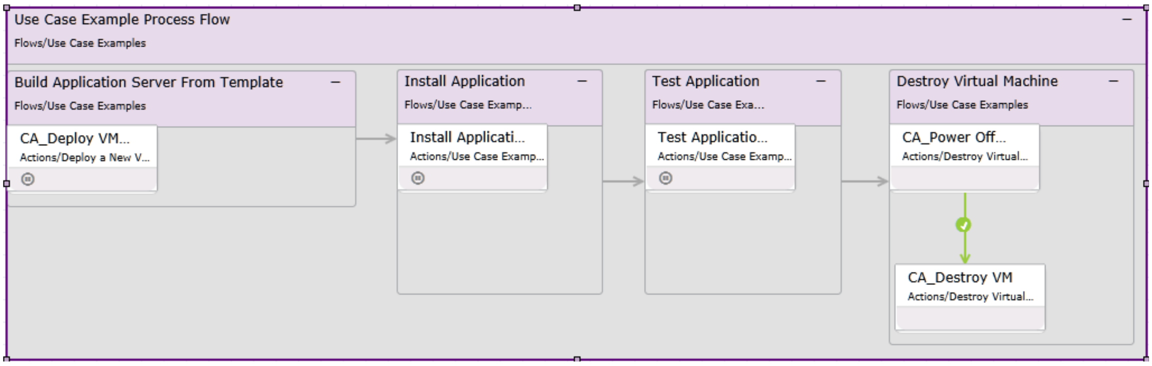
Shared components enable users to create deployment logic once and share the logic across multiple applications. Each shared component has multiple revisions with each revision acting like a regular component. Shared components help organizations establish best practices by defining standards regarding common deployments.

Using shared components ensures that all teams follow the same guidelines and workflows. For example, if you have multiple applications that use a Tomcat server, define the flows in a shared component and use it in all applications.

Introduction

This document focuses on Shared Components built from the VMware vSphere Action Pack which automate numerous VMware related activities.

CA has created a library of Shared Components that give users a quick set of Workflows to select from as they build Processes within their Release Automation infrastructure. The following is a quick example of a Process that is built using four (4) Shared Components.



In this example a shared component that creates a virtual machine is added to the beginning of the process. Next another Shared Component that installs a particular Application is added. Assuming this process is being used in the Development or QA Environments, a third Shared Component that is designed to run the application through a variety of test is added to the process. Finally, to complete the process, once all of the testing is performed, the last shared component added to the process turns off the virtual machine and destroys it (deletes it from the environment).

Understanding VMware vSphere Actions

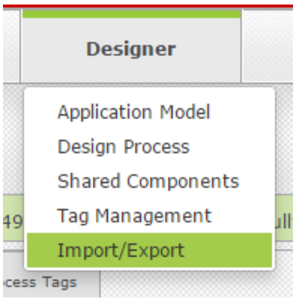
There are over 70 VMware vSphere Actions that can be grouped into three basic categories of functionality:

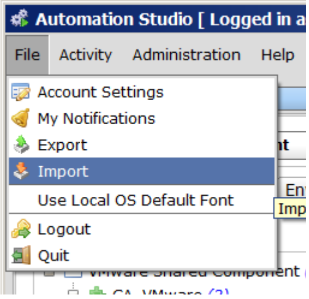
- Actions that build and create Virtual Machines
- Actions that manage and configure an existing Virtual Machine
- Actions that manage and configure the VMware Infrastructure

From a Shared Component perspective, we focused our initial attention on assisting customers with building, managing, and configuring their user’s virtual machines and not the VMware infrastructure.

Installation of VMware vSphere Shared Components

To install the VMware vSphere Shared Components you must Import the VMware_CA.zip file. In CA Release Automation Version 5.5.1 and higher this is done through the **Designer > Import/Export**. In CA Release Automation Versions 5.5 this is done through the **Automation Studio File > Import**.

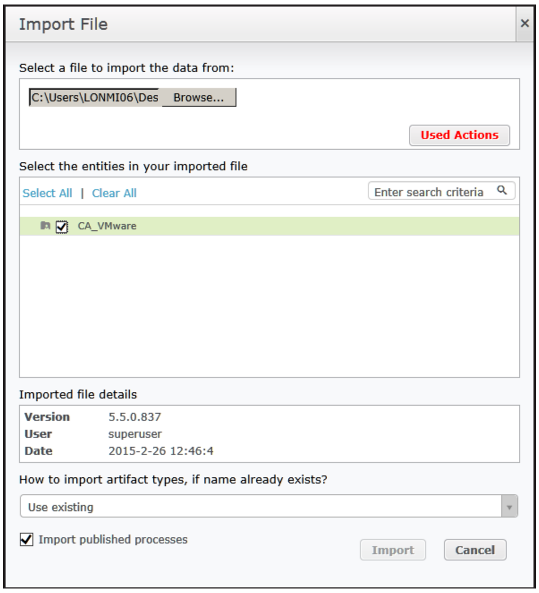




Once you are in the Import File interface, you will need to locate the **CA_VMware_V0.1zip** file that contains the VMware vSphere Shared Components.

Note: The **nolio-actions-vmware** must be already loaded into Action Management

Next, check the box next to CA_VMware and then the Import button at the bottom of the interface.



Select a file to import the data from:

C:\Users\LONMI06\Des

Select the entities in your imported file

☑ CA_VMware

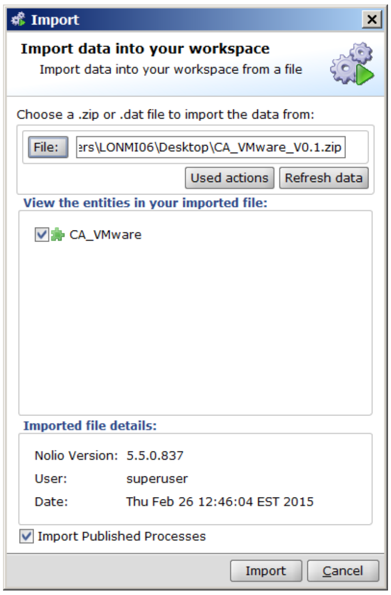
Imported file details

Version	5.5.0.837
User	superuser
Date	2015-2-26 12:46:4

How to import artifact types, if name already exists?

Use existing

☑ Import published processes


Import
Import data into your workspace
Import data into your workspace from a file
File:
Used actions
Refresh data
Import
Cancel

Choose a .zip or .dat file to import the data from:

File: ers\LONMI06\Desktop\CA_VMware_V0.1.zip

View the entities in your imported file:

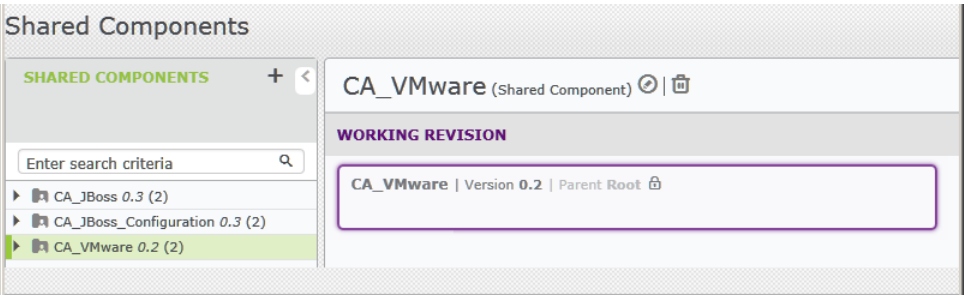
☑ CA_VMware

Imported file details:

Nolio Version:	5.5.0.837
User:	superuser
Date:	Thu Feb 26 12:46:04 EST 2015

☑ Import Published Processes

Once the Shared Components are imported go to Shared Components and confirm.



What's New

This is the initial release of the VMware vSphere Shared Components that are being supplied by CA Technologies.

Fixes

Initial Release

Workflows

How the VMware vSphere Actions Work

All VMware vSphere Actions must communicate and execute their action through the VMware vCenter Server and it is the VMware vCenter Server that actually performs the action. The CA Release Automation Agent only executes the commands to the VMware vCenter Server.

Note: The CA Release Automation Agent does not have to be installed on the VMware vCenter Server. From a CA Release Automation perspective only one Agent is needed to perform any and all VMware related tasks.

That being said, every VMware vSphere Action has five (5) Input Values that relate directly to communication with the VMware vCenter Server:

• Server:	Hostname or IP Address of the VMware vCenter Server
• Protocol:	Protocol for vCenter communication, either http or https (Default https)
• Port:	Port Number for vCenter communication (Default value 443)
• Username:	VMware vCenter User Account with rights to perform these tasks
• Password:	VMware vCenter User Account Password

To simplify entry of all of the Input Values for each VMware vSphere Action the Input Values have been replaced with specific CA Release Automation Parameters. For example within every VMware vSphere Action these five (5) Input Values have the following Parameters entered in their Shared Components:

Input Value	Type	Parameter
Server	String	vcServer_CA
Protocol	String	vcProtocol_CA
Port	Integer	vcPort_CA
Username	String	vcUsername_CA
Password	Password	vcPassword_CA

By entering the Input Value in the Parameter, the information is then used by every VMware vSphere Action, allowing perfect communication with the VMware vCenter Server. Likewise, the use of Parameters is consistent throughout the entire setup and configuration of Shared Components, making the building of Processes with Shared Components simple, secure, and standardized in every environment where they are used.

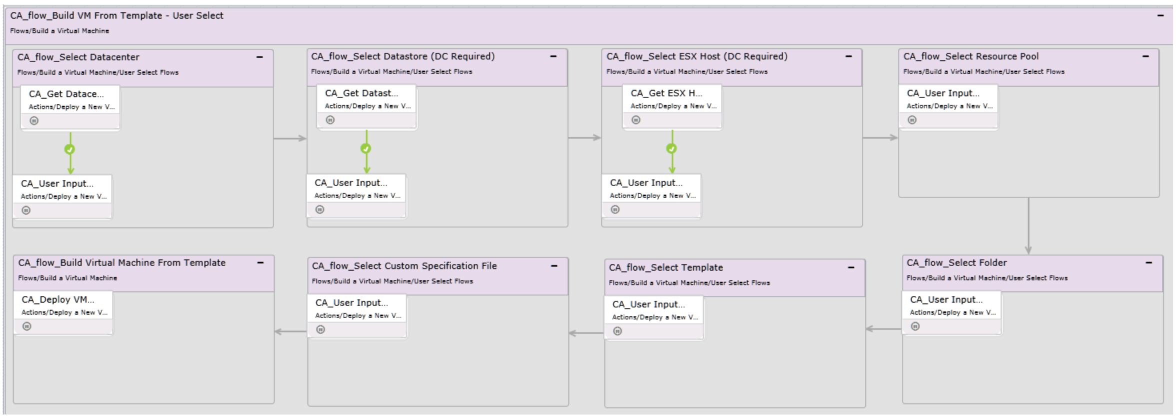
Building a Virtual Machine

One of the main uses of VMware is to build virtual machines. The VMware vSphere Actions allow you to automate this task and (as explained in the **Introduction**) simplify it down to a single Shared Component. The following VMware vSphere Actions have been preconfigured as Shared Components:

- **CA_Deploy VM From Template** – Creates a new virtual machine from an existing Template
- **CA_Clone VM** – Creates a new virtual machine that is a copy of an existing virtual machine

As mentioned earlier all of these Actions have been preconfigured with Parameters to allow you to enter the Input Value once or as needed depending upon your environment.

Additionally, in sub-folders are different sets of pre-built Shared Components that are designed for testing and gather User input. Many of them take advantage of the VMware vSphere Actions that "GET" information from vCenter, providing the User with a List of choices to select. It is expected that most organizations will have more defined VMware Infrastructures and procedures but the following is an example of a process built using this User Input


CA_flow_Build VM From Template - User Select
Flows/Build a Virtual Machine
CA_flow_Select Datacenter
Flows/Build a Virtual Machine/User Select Flows
CA_Get Datace...
Actions/Deploy a New V...
CA_User Input...
Actions/Deploy a New V...
CA_flow_Select Datastore (DC Required)
Flows/Build a Virtual Machine/User Select Flows
CA_Get Datast...
Actions/Deploy a New V...
CA_User Input...
Actions/Deploy a New V...
CA_flow_Select ESX Host (DC Required)
Flows/Build a Virtual Machine/User Select Flows
CA_Get ESX H...
Actions/Deploy a New V...
CA_User Input...
Actions/Deploy a New V...
CA_flow_Select Resource Pool
Flows/Build a Virtual Machine/User Select Flows
CA_User Input...
Actions/Deploy a New V...
CA_flow_Select Folder
Flows/Build a Virtual Machine/User Select Flows
CA_User Input...
Actions/Deploy a New V...
CA_flow_Select Template
Flows/Build a Virtual Machine/User Select Flows
CA_User Input...
Actions/Deploy a New V...
CA_flow_Select Custom Specification File
Flows/Build a Virtual Machine/User Select Flows
CA_User Input...
Actions/Deploy a New V...
CA_flow_Build Virtual Machine From Template
Flows/Build a Virtual Machine
CA_Deploy VM...
Actions/Deploy a New V...

The following is a suggested way of using either of these Shared Components to simplify and quickly configure them for use within your environment:

1. Validate and get the unique names of the VMware Templates and Custom Specification Files that you will be using to deploy virtual machines. For example; you may have a standard Windows and Linux Template that are used as well as Custom Specification Files for each.
2. Depending upon how you use the Shared Component in reality you will probably want to enter the value of many of the Parameters, especially those that are consistent throughout.
3. Make a copy of the **CA_DeployVM** Shared Component for each type of virtual machine you wish to deploy and define the unique Parameters in each copy (VMware Template and Custom Spec File). For example:
   - "***CA_DeployVM_Windows***" – All Parameters would be identical except this Component would have a unique Windows Template and Custom Spec File defined.
   - "***CA_DeployVM_Linux***" – Likewise, this Component would have all Parameters identical except a unique Linux Template and Custom Spec File would be defined.

Managing a Virtual Machine

Once a user has a virtual machine there are numerous things that they might want to do to it. They may need to change the Power State, turn the VM off, on, or reboot the server. They may want to take Snapshots and of course delete and revert to a particular Snapshot. They may need to adjust the hardware configuration, adding a second drive, CD, ISO, NIC, or change the memory, and they may need to delete the virtual machine once they are done with it. The following are the VMware Shared Components that are currently available for this use:

Shared Components to Delete a Virtual Machine

- **CA_Destroy VM** – will delete a virtual machine that is Powered Off
- **CA_flow_VMware_destoryVM** – this flow will first Power Off the VM (but this Action does not have to succeed in case it is already Powered Off, and then next the flow performs the CA_Destroy VM Action.

Shared Components to Change the Power State of a Virtual Machine

- **CA_Power Off VM** – will Power Off the VM
- **CA_Power On VM** – will Power On the VM
- **CA_Reset VM** – will reset a specific VM
- **CA_Resume VM** – will start a “Suspended” VM
- **CA_Shutdown VM** – will perform a graceful system Shutdown of a VM
- **CA_Suspend VM** – will put a VM in a suspended state
- **CA_flow_VMware_rebootVM** – there is no reboot action so this flow accomplishes the same task, it first perform a CA_Shutdown VM bringing the VM down gracefully and then perform a CA_Power On VM to bring the VM back up and operational.
- **CA_flow_VMware_supsendPowerOff** – this flow is used when a VM is in a “Suspended” state and you wish to Power it off. The VM is first made operational again by the CA_Resume VM action and then powered off by the CA_Power Off VM action.

Shared Components to Manage Snapshots

- **CA_Create Snapshot** – will create a Snapshot for a specific VM
- **CA_Remove All Snapshots** – will remove all of the Snapshots on a VM
- **CA_Remove Snapshot** – will remove a specific Snapshot from a VM
- **CA_Rename Snapshot** – will rename a specific Snapshot from a VM
- **CA_Revert To Snapshot** – reverts a VM to a specific Snapshot. It should be noted that after a VM is reverted it is in a Powered Off state.
- **CA_flow_VMware_revertSnapshotPowerOn** – this flow uses the CA_Revert To Snapshot action which reverts a VM to a specific Snapshot and then it powers the VM back on using the CA_Power On VM action.

Environment Variables

As mentioned earlier every VMware vSphere Actions and thus all of the Shared Components use the VMware vCenter Server to execute the actions. For this reason, every Shared Component has these Parameters:

• **vcServer_CA**	vCenter Server (Hostname or IP Address)
• **vcProtocol_CA**	vCenter Server Protocol (default https)
• **vcPort_CA**	vCenter Server Port (default 443)
• **vcUsername_CA**	vCenter Server Username
• **vcPassword_CA**	vCenter Server User Password

Building a Virtual Machine

The following is a breakdown of the Input Value or Parameters that need to be entered for each of these two Shared Components that are used to build virtual machines:

• **vcTMP_CA**	VMware Template Name to use to deploy new VM from
• **vcFolderPath_CA**	Full Path to VMware Template (default Root folder)
• **vcName_CA**	Name of Virtual Machine
• **vcNewFolderPath_CA**	Full Path to new VM (default Root folder)
• **vcDatacenter_CA**	VMware Datacenter where new VM will be located
• **vcHost_CA**	VMware ESX where new VM will be located
• **vcResourcePool_CA**	VMware Resource Pool where new VM will be located
• **vcDatastore_CA**	VMware Datastore where new VM will be located
• **vcPowerOnAuto_CA**	Power On VM Automatically (Parameter set to “true”)
• **vcCustomSpec_CA**	VMware Custom Specification File to use to build new VM
• **vcTimeout_CA**	Timeout value to use to determine tasks has failed (default 900)

Managing a Virtual Machine

Again the VMware vCenter Server Parameters are used on all of these Shared Components. In fact those five Parameters plus the Virtual Machine’s Name and the Timeout are many times all that is required to perform many of these tasks.

• **vcServer_CA**	vCenter Server (Hostname or IP Address)
• **vcProtocol_CA**	vCenter Server Protocol (default https)
• **vcPort_CA**	vCenter Server Port (default 443)
• **vcUsername_CA**	vCenter Server Username
• **vcPassword_CA**	vCenter Server User Password
• **vcName_CA**	Name of Virtual Machine
• **vcTimeout_CA**	Timeout value (default 900)

The following are some of the other Parameters used by some of the other Shared Components:

• **vcRetries_CA**	Number of retries to attempt
• **vmSnapshotName_CA**	Name of Snapshot
• **vmNewSnapshotName_CA**	New Name for Snapshot
• **vmDescription_CA**	Description for Snapshot