

Release Automation 6.3 & DevTest Integration

Manjunath Narayan
Engineering Services Architect
SWAT-Advocate Team

Scope

Demonstrate the Release Automation Integration with DevTest

Integration of Release Automation and DevTest enables you to perform Continuous Delivery and Continuous Testing which will help you to identify the defects early on the lifecycle and implement DevOps Best Practices.

DevTest Features

CA Application Test

- Provides an automated testing solution
- design and execute automated tests for multiple layers of a distributed architecture.

CA Service Virtualization

- Removes constraints throughout the SDLC by modeling and simulating unavailable or dependent systems.
- Reduces demand for lab infrastructure and software to avoid costs and reduce configuration effort.

RA integration with DevTest

Procedure

- Install the RA Agent on the Dev Test Server
- Install the DevTest Action Pack
- Adding Action for Service Virtualization
- Adding Action for Application Test
- Sample Flow

Install the RA Agent on the Dev Test Server

From ASAP

The screenshot shows the Automation Studio Administration console. The left sidebar has 'Agent Management' highlighted. The main area shows 'Agent Groups' with a list of agents. A red box highlights the 'Install Agent on a Windows server' option. An 'Agent Installation' dialog box is open, titled 'Install new Agent'. It contains the following fields:

- Server Names: *separate different agents with ',' (narma13-u187510)
- User Name: Administrator
- User Password: [masked]
- Installation Directory: C:\Program Files\CA\ReleaseAutomationAgent
- Agent TCP Port: 6600
- Execution Server: narma13-U182331
- Execution Server name or IP: narma13-U182331

At the bottom of the dialog, there are 'Advanced...' and 'Install' buttons. To the right, an 'Advanced properties' dialog is partially visible, showing 'Service User name' as localhost\Administrator and 'Service password' as [masked]. A 'Save and Close' button is also present.

Install the RA Agent on the Dev Test Server

From Release Automation (<http://RAserver:8080/nolio-app/>)

ca technologies

Explore Release Planning & Management
Free license to try CA Release Automation
Continuous Delivery Edition [Check it out >](#)

CA Release Automation

Managing enterprise application release operations

Release Operations Center

Central hub to plan, design, execute and track deployments

Launch

Automation Studio

Legacy IDE for administration and application deployment modeling
(Automation Studio requires Oracle JRE™ 8 and runs using Java™ Web Start Launcher)

Launch

Download Agent Setup: Agent Setup (Windows) Download

- Agent Setup (Windows)
- Agent Setup (Linux x64)
- Agent Setup (Solaris-SPARC)
- Agent Setup (Solaris-x86)
- Agent Setup (AIX)
- Agent Setup (Solaris 9 SPARC)
- Agent Setup (Solaris 9 x86)

Installation of Action Pack (nolio-ca-lisa-actions)

CA Release Automation Dashboard Designer Releases Artifacts Environments **Administration** DSL Switch To Express Mode | superuser | Help

Action Management

66* Available Packs
66 new packs available

8* Installed Packs
One updated pack available

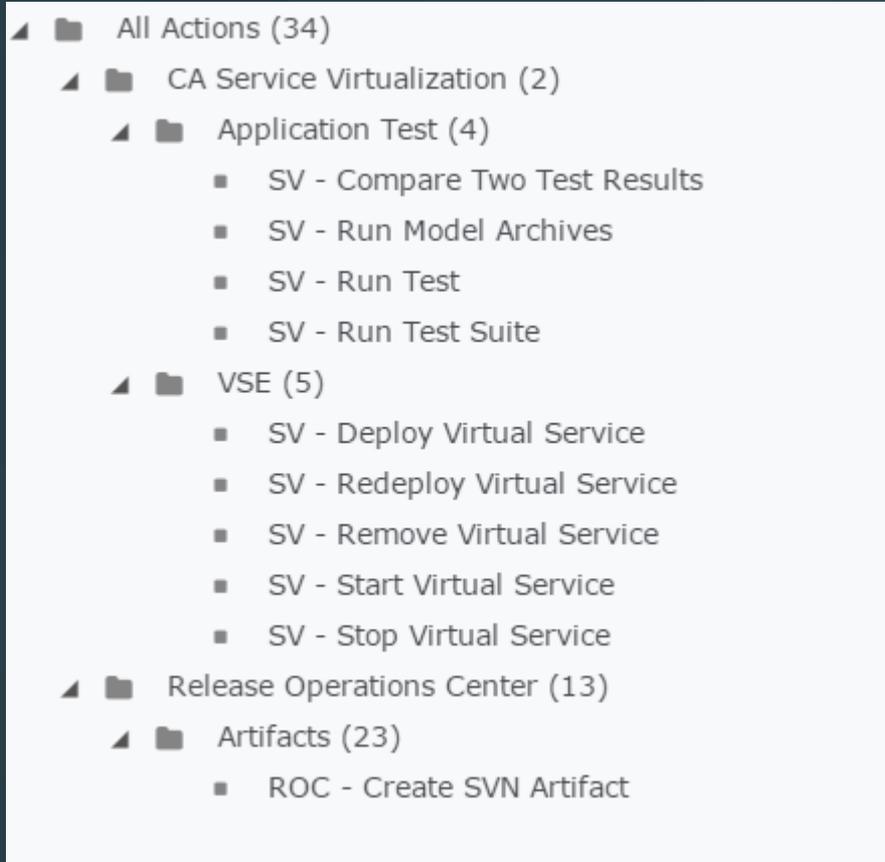
Enter search criteria

Refresh Download

<< < 1 - 10 of 66 > >>

<input type="checkbox"/>	Available Packs	Description	Action List	Version	Date
<input type="checkbox"/>	nolio-actions-f5bigip ☆	Nolio F5 Big IP suite actions package		10.5.1410.2	2016-07-26
<input type="checkbox"/>	nolio-actions-vmware ☆	Nolio VMWare actions package		10.4.1506.1	2016-04-20
<input type="checkbox"/>	nolio-amazon-actions ☆	Nolio Amazon actions package		10.4.1506.1	2015-07-15
<input type="checkbox"/>	nolio-atlassian-jira-actions ☆	Nolio JIRA actions package		10.7.1506.1	2015-07-15
<input type="checkbox"/>	nolio-biztalk-actions ☆	Nolio Biztalk actions package		10.4.1412.1	2014-12-18
<input type="checkbox"/>	nolio-ca-harvest-actions ☆	Nolio CA Harvest actions pack		10.3.1503.1	2015-03-19
<input checked="" type="checkbox"/>	nolio-ca-lisa-actions ☆	Nolio CA Service Virtualization actions package		10.9.1504.1	2016-07-31
<input type="checkbox"/>	nolio-ca-servicedesk-actions ☆	Nolio CA ServiceDesk actions pack		10.3.1402.1	2014-03-26
<input type="checkbox"/>	nolio-chef-actions ☆	Nolio Chef actions pack		10.4.1406.1	2016-11-24
<input type="checkbox"/>	nolio-clearcase-actions ☆	Nolio clearcase actions pack		10.3.1411.1	2014-11-04

Installation of Action Pack (nolio-ca-lisa-actions)



Creating the Action - Deploy Virtual Service

The screenshot displays the Camunda BPMN editor interface. On the left, the 'Processes' tab is active, showing a tree view of components. The 'CarService - Deploy Virtual Service' action is highlighted, and a context menu is open with 'Add Action' selected. Below this, the 'Add Action' dialog is shown with 'deploy vi' entered in the search field. The 'All Actions' list shows 'SV - Deploy Virtual Service' selected. The main editor area shows the configuration for the 'CarService - Deploy Virtual Service' action, including its name, description, and various input fields.

Components | **Processes**

Enter search criteria

- Default Component (2)
 - Actions (7)
 - TomCatDeploy (6)
 - CarService - Deploy Virtual Service**
 - CarSe...
 - Get W...
 - SV - Dv...
 - SV - Red...
 - SV - Ra...
 - Flows (1) st...

Actions Category Options

- Add Action**
- New Category
- Copy

Add Action

deploy vi

All Actions

- All Actions (34)
 - CA Service Virtualization (2)
 - VSE (5)
 - SV - Deploy Virtual Service**
 - SV - Redeploy Virtual Service

CarService - Deploy Virtual Service (action) || type: SV - Deploy Virtual Service

Name

CarService - Deploy Virtual Service

Deploy a mar file to VSE as virtual service. (SV version 7.5.1 or higher)

Inputs	Outputs
SV Server Name [String]	
narma13-u187510	
SV Invoke Port [int]	
1505	
SV User Name [String]	
admin	
SV User Password [Password]	
.....	
Model Archive (MAR) PATH [String]	
C:\Program Files\CA\DevTest\Projects\Cars-Demo1\VirtualServices\CarService.mar	
Virtual Service Environment [String]	
VSE	

Creating the Action - Stop Virtual Service

CarService - Stop Virtual Service (action) || type: SV - Stop Virtual Service

Name

CarService - Stop Virtual Service

Stop a virtual service in a VSE. (SV version 7.5.1 or higher)

Inputs	Outputs
SV Server Name [String]	
narma13-u187510	
SV Invoke Port [int]	
1505	
SV User Name [String]	
admin	
SV User Password [Password]	
••••••	
Virtual Service Name [String]	
CarService	
Virtual Service Environment [String]	
VSE	

Settings for running the Test Case

```
## -----  
## IP Spoofing  
##  
## Interfaces: a comma-separated list  
## MAC address (JDK 1.6+), interface n  
##  
#lisa.ipspoofing.interfaces=12-34-56-7  
  
## -----  
## Force the use of a native file dial  
#lisa.ui.useNativeFileDialog=true  
  
## -----  
## Specify the maximum number of recen  
## in the Open Recent Quick Start tab  
#lisa.prefill.recent=10  
#lisa.quickstart.recent=5  
  
##=====lisa-invoke=====  
#lisa.portal.invoke.base.url=/lisa-inv  
#lisa.portal.invoke.report.url=/report  
#lisa.portal.invoke.server.report.dire  
#lisa.portal.invoke.test.root=d:/lisat  
  
##=====DevTest Integration Kit example  
#asserts=com.mycompany.lisa.AssertFile  
#com.mycompany.lisa.AssertFileStartsWi  
#filters=com.mycompany.lisa.FilterFileFirstLine  
#com.mycompany.lisa.FilterFileFirstLine=com.itko.lisa.editor.FilterController,com.itko.lisa.edito  
#nodes=com.mycompany.lisa.node.FTPTestNode  
#com.mycompany.lisa.node.FTPTestNode=com.mycompany.lisa.node.FTPTestNodeController,com.mycompany.  
  
# -----  
# This property is used as backlog argument while binding Server Socket. Backlog is the requested  
# Its exact semantics are implementation specific. In particular, an implementation may impose a  
# The value provided should be greater than 0. If it is less than or equal to 0, then by default  
# -----  
#lisa.serverSocket.bind.backlog=100  
  
# -----  
#lisa.portal.invoke.test.root={{LISA_HOME}}
```

The screenshot shows a Windows File Explorer window with the address bar containing the path: This PC > Iodisk1 (C:) > Program Files > CA > DevTest. The file list below shows various files and folders. The file 'local.properties' is highlighted with a red box. Other files in the list include '_res-hub-config.properties', '_selenium-config.json', '_site.properties', 'appletviewer.policy', 'configure.bat', 'contenttypes.xml', 'de-identify.xml', 'derby.log', 'Devtestlic.xml', 'devtestlic-old.xml', 'httpbrowsers.xml', 'ims.format', 'j2eeservers.xml', 'lisa.permissions', 'lisa.properties', 'logging.properties', and 'mibdefs.xml'.

Creating the Action – Run Test Suite

SV - Run Test Suite (action) || type: SV - Run Test Suite

Name

SV - Run Test Suite

This action runs test from suite file (*.ste) synchronously. User should set the **root** property `lisa.portal.invoke.test.root={{LISA_HOME}}` in **lisa.properties** file before run the action. The paths of files required by this action are relative to the **root**

Inputs

Outputs

Settings

SV Server Name [String]

narma13-u187510



SV Invoke Port [int]

1505

Test Suite Path [String]

examples/Suites/AllTestsSuite.ste



Configuration Path [String]

No Value



Fail If Any Case Failed [boolean]

Default value [false]

Demonstration

Reference:

Release Automation integration with CA Service Virtualization

<https://docops.ca.com/ca-release-automation/integrations/en/optional-action-packs/ca-service-virtualization>

DevTest

<https://docops.ca.com/devtest-solutions/9-5/en>



Manjunath Narayan

Enterprise Service Architect

narma13@ca.com



in