



Software version: 1.80 For Microsoft Windows® and Linux operating systems

Codar Plugin Automation for Continuous Integration Tool

Document release date: January 2016 Software release date: January 2016

Contents

Introduction	3
Types of designs	3
Codar Topology Design	3
Codar Sequence Design	4
Integration Automation	7
Sample Implementation	
Conclusion1	1
Send documentation feedback1	
Legal notices1	2

Introduction

Codar is a continuous delivery automation tool which supports deployment steps and pipeline process automation. It is always the choice for the customer to integrate the CI tool with continuous delivery tool like Codar in order to automate the process of CI-CD.

Any code which is built by the build tool should be first deployed and verified on the dedicated environment before it is consumed by other lifecycle stages. This will help the team to regress the code further and finally rollout to production.

The content of this whitepaper will expose the API's which are really required to connect or integrate CI with Codar tool.

Also the steps to automate the integration process are documented in this whitepaper. For any more details on the product features please refer the product guides.

Types of designs

There are two types of designs which are supported by Codar which are Topology and Sequence. This section will explain the list of API which are to be used to automate the integration of CI tool with Codar's topology design and Codar's sequence design.

Codar Topology Design

Following are the information which are required to automate the integrate,

- 1. Application design id
- 2. Component Id
- 3. Application JSON to fetch the component property names which are modifiable by CI tools
- 4. Create package API or flow
- 5. Promote package API or flow with continuous promote yes or no option

There are multiple ways to get the above information. One of the easiest way is to export the topology design as JSON and the id can be fetched from it. The other option is to access swagger API portal to get the list of designs which will have all the design id with JSON body. In this topology example we are going to get the information from topology JSON and for sequence design we will see how to fetch the same information through API in the section <TODO>

Steps to fetch the information which are required to integration topology application design with CI tool

- 1. Login to Codar
- 2. Go to Designs → Topology → Designer → Search (type the application name you want deploy through CI-CD integration)
- 3. Go the specific version of the application design
- 4. On the gear box 🖻 click on the Export
- 5. This will download a generated JSON file
- 6. Open the JSON file which is of below format and the highlighted string is the id of this design

```
{
   "@self" : "/csa/api/topology-model/topology/b81bab25-c2ef-4fc3-88f3-cb8cb6e916d1",
   "@type" : "urn:x-hp:2013:software:cloud:topology_model:topology",
   "groupId" : "com.hp.csa",
   "artifactId" : "c805f5de5e5447d9b1a262a8f445dd29",
   "version" : "5.0.0",
   "displayName" : "PetClinic Application - Partial Design - Release",
   "description" : "A two-tier PetClinic Application with Database component installed on MySQL
```

7. In order to get the component id which contains the parameters which are modifiable during deployor re-deploy, scroll down to look for the component names which will give the detailed information about the component and id.

In the sample JSON the component name "PetClinic Application 4 Partial Design' has the modifiable property which is "artifacturl". Search for this component name and fetch the id highlighted as given below,

```
requirements : [ ]
}, {
    "id" : "9b4da342-8b49-8c5d-1db4-8062778c9e3h",
    "name" : "PetClinic Application 4 Partial Design",
    "component" : {
        "@self" : "/csa/api/topology-model/component-type/a22516ee-ff82-4686-ad10-33056421f08f",
        "groupId" : "com.hp.csa.type.HPOO",
        "artifactId" : "PetClinicApplication4PartialDesign_bfccdc9dfdf14de788131fd7fdb80f0f",
        "version" : "1.50.0000"
},
```

Property which is modifiable during deployor re-deploy which is present as part of this component. The value of this property should be set by the CI tool

```
}
}, {
    "propertyKey" : "artifacturl",
    "propertyValue" : {
```

8. By following the above step you should have application design id, application component id and the property which are required to be modified from CI tool.

Codar Sequence Design

As mentioned in the previous section, it will require APIs to provide all the required information to automate the integration between CI tools with Codar's sequence design.

Steps to fetch the information which are required to integration sequence application design with CI tool

- 1. Login to Codar
- 2. Attach the following URI with the logged-in URL "apidocs.jsp" so the URL should be as given below

https://Codarmc:8444/csa/apidocs.jsp#!/

Go to the section "sequence: The API for managing service design containers. (internal use only)"

sequence : The API for managing service design containers. (internal use only)									
	Show/Hide List Operations Expand Operations Raw								
POST /container/sequence/filter	Query for service designs matching a filter on tag								
GET /container/sequence/	Returns a list of all existing service design containers								

3. Click on the "Try it out" button on the subsection. This will list all the sequence design along with the JSON body.

GET /contair	ner/sequence/		Returns a list of	all existing service design containers
Implementation The API respons	n Notes se returns the collection of containers for service of	lesigns existing in the system.		
Parameters				
Parameter	Value	Description	Parameter Type	Data Type
start-index		Specifies the offset of the first entry to be included in the page.	query	integer
page-size		Specifies the page size.	query	integer
sort		Name of field to be used in ordering optionally followed by colon and 'ascending' or 'descending'	query	string
after		Filter members to those modified at or after this timestamp. Uses SimpleDateFormat("yyyy-MM- ddT'HH:mm:ss.SSS'Z"") in UTC	query	string
before		Filter members to those modified before this timestamp. Uses SimpleDateFormat("yyyy-MM- ddT'HH:mm:ss.SSS'Z"') in UTC	query	string
Error Status Co	des			
HTTP Status Cod	e Reason			
404	No containers found.			
403 Try it out!	Authorization failure			

4. The below JSON body will contains the information about the application container id as well as the versions under this container. The members section will give the version id of the sequence design. In the below screen shot the id's which are inside the box are application version design id.

```
ſ
  "@self": "/csa/api/container/sequence/f5310bb1135845c7b0374f85980e62a4",
  "@type": "urn:x-hp:2012:software:cloud:data model:blueprint:collection",
  "@created": "2016-09-23T13:41:37.397Z",
  "@modified": "2016-09-23T13:52:36.581Z",
  "global id": "f5310bb1135845c7b0374f85980e62a4",
  "name": "vCenter Compute with Basic Options",
 "members": [
   {
              "/csa/api/service/design/9cb7cdbb002442c7959771daf0fa2b27",
     "@self":
     "@type": "urn:x-hp:2012:software:cloud:data model:blueprint"
     "@created": "2016-09-23T13:41:38.198Z",
     "published": false,
     "version": "16.07b",
     "upgrade available": false,
     "designId": "9cb7cdbb002442c7959771daf0fa2b27"
   },
   {
     "@self": "/csa/api/service/design/3afb4361e79f4caca092e90aa1c18cfa",
     "@type": "urn:x-hp:2012:software:cloud:data model:blueprint",
     "@created": "2016-09-23T13:52:36.581Z",
     "published": false,
     "version": "16.07",
     "upgrade_available": false,
     "designId": "3afb4361e79f4caca092e90aa1c18cfa"
   }
```

5. Now the next step is to fetch the component id(s) present inside application version JSON body. This information also can be fetched from the swagger API available as part of Codar.

app-package : The API to Manage Packages

Show/Hide List Operations Expand Operations Raw

6. Open the API "GET /codar/app-package/{applicationDesignId}/designComponents" and provide the application version design id to fetch the component information by clicking "Try it out" button

GET /codar/app-p	oackage/{applicationDesignId}/designCompon	ents		Get all the design components
Response Class Model Model Schema	а			
Map { empty (boolean, optio	nal)			
<pre>}</pre>	iraij			
Response Content Ty	pe application/json ▼			
Parameters				
Parameter	Value	Description	Parameter Type	Data Type
applicationDesignId	9cb7cdbb002442c7959771daf0fa2b27	The id of the application design	path	string
Error Status Codes				
HTTP Status Code	Reason			
400	Bad request			
401	Authorization failure			
404	Not found			
500	Internal server error			
Try it out! Hide Res	<u>ponse</u>			

7. The component id can be fetched from the response body (JSON output) which is show from the above API after "Try it out!"

The highlighted id is the component id which should be used to pass any input to the property which is present as part of the component. The "displayName" is the property name which is highlighted in the second box in the below screen shot



Note: Sequence based design Codar does support all properties as modifiable but re-deploy is not possible.

Integration Automation

Here is the master piece which is going to integrate CI tool with Codar. Codar has got an HPE Operation Orchestration (HPE OO) flow which can automatically create a package and do a continuous promote release pipeline or mere deployment on first lifecycle stage.

The OO flow name and the path is "Library/Integrations/Hewlett-Packard/Cloud Service Automation/Components/CODAR/Devops/Continuous Deployment Flow.xml"

The OO flow can be triggered remotely from a command line tool called "RSFlowInvoke.exe" with the required options.

On Linux environment JRSFlowInvoke.jar can be used to trigger the workflow with the same option.

RSFlowInvoke.exe and JRSFlowInvoke.jar is supported only for HPE OO Central 9.x version but this tools still works with 10.x version.

HPE OO 10.x comes with the tool called "OOSH.bat/OOSH.sh" to accomplish the same task of invoking or triggering the OO flows remotely from a CIT tool. This tool may be further enhanced in the next version which can be invoked as standalone tool.

This flow will take the following input in order to remotely trigger an OO flow,

RSFlowInvoke.exe -host <CODARHOST>:<CODARPORT> -flow 0866af7f-568a-4d73-bd55-5be734aa7d15 -a Basic -u <OO Central Server Username> -p <OO Central Server Password> -t <Flow Timeout> -inputs "<Input to the application flow>" **Sample Command**

RSFlowInvoke.exe -host16.103.31.119:8445 -flow 0866af7f-568a-4d73-bd55-5be734aa7d15 -a Basic -u admin -p cloud -t 600 -inputs "codarusername=codaruser&codarpassword=admin@123&designurl=null&package=508a6ff1c3ed4168a97d97c33c756549=[{Number of

Servers:1}]&applicationDesignId=87153c7ce689466cbccd8ebb79964d17&buildId=Package1&packageName=Package1&continuousPr omote=true&description=This is triggered from bamboo"

In the above command the option input has the following values,

Continuous Deployme	ent Flow	
ID:	0866af7f-568a-4d73-bd55-5b	e734aa7d15
Content Pack:	CODAR	
Description:		
This flow can be usd wh		Continuous Integration (CI) and Continuous Deployment. A eapplication is deployed using model on a specific environment
Run Name		Continuous Deployment Flow
Persistence Level		Standard
designurl:		k null
package:		508a6ff1c3ed4168a97d97c33c756549=[{Number of Servers:1}]
applicationDesignId:		87153c7ceó894óócbccd8ebb799ó4d17
buildId:		Package01
packageName:		Package01
httpusername:		
httppassword:		
environment:		
continuousPromote:		frue
description:		This is triggered from Bamboo

The password can be encrypted using the same tool and passed with an option "-ep"

The SDK guide explain much more details about this tool with all the options.

This flow can be invoked from any CI tool with any of the native plugin.

Sample Implementation

In this section we are going to show how a Codar plugin can be created for a CI tool using the native features of the CI tool.

The sample tool taken is Bamboo. Please note that Codar already has a bamboo plugin to trigger topology based design with continuous promote yes or no option. In this sample we are going to see how a sequence design can be used in a pipeline to stand up an infrastructure, along with that install or configure the software and deploy the application remotely and how a pipeline can be kick started from Bamboo.

For more details on the Bamboo CI tool please refer the Bamboo help guide.

In the above or previous section we say how a flow can remotely invoked using a tool called "RSFlowInvoke.exe".

We will now see how this tool can be integrated with Bamboo and start an application release pipeline by passing appropriate inputs.

Step to add an executable within Bamboo too.

1. Create new executable in the Server capabilities as given in the below screen shot.

Bamboo My Bamboo B	uild - Deploy - Reports - Create -
Bamboo administrati	on
BUILD RESOURCES	Server capabilities
Agents	You can use this page to view, add and delete server capabilities. These capabilities will be inherited by all local agent
Agent matrix	Executable
Executables	'executable' capabilities define the executables which are available to your build plans.
JDKs	
Server capabilities	Executable label Path
Global variables	A label to uniquely identify this executable Please en

2. Create a new capability

Add capability		
Capability type	Executable	•
Туре	Command	
Executable label		
	A label to uniquely identify this executable	
Path		
	Please enter the path to your executable	
	Add Cancel	

3. Copy the "RSFlowInvoke.exe" tool on the Bamboo system and place it under a directory. The "Path" should have the value as given below in the screen shot which is the actual path of the RSFlowInvoke.exe which is present on the Bamboo system.

Capability details

Shared capabilities	Server capabilities						
Capability type	Executable						
Executable label	Deployment Request						
Path	C:\RSWorkflow\RSFlowInvoke.exe						
	Please enter the path to your executable						
	Update Cancel						

4. Create a "new build plan"

Bamboo My Bamboo	o Build -	Deploy -	Reports -	Create -	
Build projects Provision Se	erver			Create a new plan Clone an existing plan Create deployment pro	ject

5. Click on the "default job"

Build projects / Provision Server / Provision Server Test Configuration - Provision Server Test									
Plan Configuration									
Stages & jobs	1	Pla	n details	Stages	Repositories	Т			
Branches	0	Plan contents							
		Each stage within a plan represents a step with followed by a stage for deployment jobs.							
		III De	fault Sta	ge					
		iii Default Job							
		+ Ad	ld job						

6. Add Task to the default job

Plan Configuration				
Stages & jobs	Job details Tasks	Requirements	Artifacts	Miscellaneous
Default Stage	Tasks			
듣 Default Job	A task is a piece of work the	nat is being execu	ited as part	rt of the build. The execution of a script, a shell command, an Ant Task or a Maven goal are only few examples
Branches 0	You can use runtime, plan	and global variab	ples to para	ameterize your tasks.
	Command		8	Command configuration
	Deploy Application Final tasks Are always execute	ed even if a provinue to	aak faila	Task description
		to make them fina		Deploy Application
				Disable this task
	Add task			Executable
				Deployment Request
				Argument
				-host 16.103.31.119:8445 -flow 0866af7f-568a-4d73-bd55-5be734aa7d15 -a
				Argument you want to pass to the command. Arguments with spaces in them must be quoted
				Environment variables
				Extra environment variables. e.g. JAVA_OPTS="-Xmx256m -Xms128m". You can add multiple parameters separated by a sp
				Working sub directory

7. In the above screen shot the executable selected is already configured executable for "RSFlowInvoke.exe"

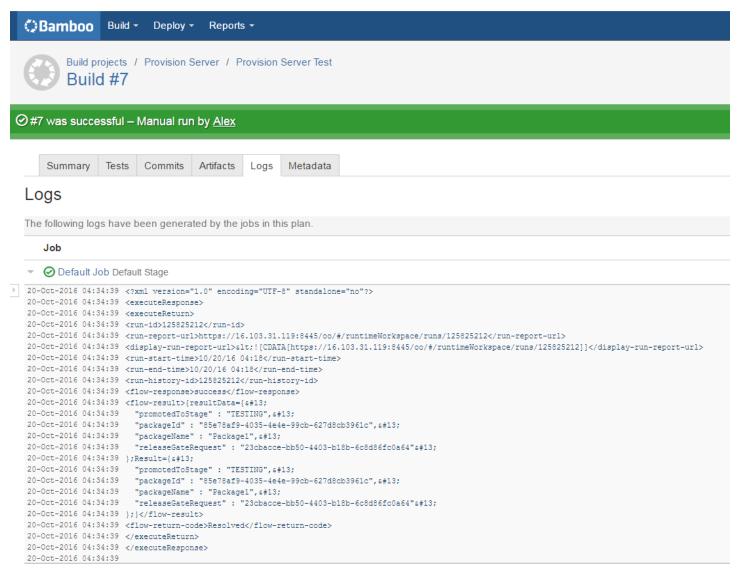
The arguments should be the RSFlowInvoke.exe arguments which is given below and explained in the "Integration Automation" section.

8. Once it is configured trigger a test run to test the configuration by clicking the highlighted item in the box

Plan Configuration Stages & jobs	1	Job details	Tasks	Requirements	Artifacts	Miscellaneous	_		Run ▼	Actions -
Default Stage		Tasks					1	Run plan		
🕨 Default Job		A task is a piece	of work tl	nat is being exect	uted as part	of the build. The execu	itic	Run custo	mised	
Branches	0	You can use runt	ime, plan	and global varial	bles to para	meterize your tasks.		Ruit custo	miseu	ļ
					2					
		Command Deploy Applica	tion		۲	Command con	nfi			

This is currently configured as part of the build task but this certainly can be configured as part deployment job as well

The below is the output of the run triggered from Bamboo plugin which invokes the OO flow to start the pipeline after creating the package in the design configured.



Conclusion

The instructions given in the whitepaper can be used to develop a quick and simple plugin for any CI tool to integrate with Codar in order to initiate the continuous pipeline after creating the package with required inputs. The API's reference in this whitepaper mayget enhanced and please follow the Codar guides which has references for all the APIs.

Send documentation feedback

If you have comments about this document, you can send them to clouddocs@hpe.com.

Legal notices

Warranty

The only w arranties for Hewlett Packard Enterprise products and services are set forthin the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional w arranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change w ithout notice.

Restricted rights legend

Confidential computer software. Valid license from Hew lett Packard Enterprise required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright notice

© Copyright 2016 Hew lett Packard Enterprise Development LP

Trademark notices

Adobe® is a trademark of Adobe Systems Incorporated.

Microsoft® and Window s® are U.S. registered trademarks of Microsoft Corporation.

Oracle and Java are registered trademarks of Oracle and/or its affiliates.

UNIX® is a registered trademark of The Open Group.

RED HAT READY™ Logo and RED HAT CERTIFIED PARTNER™ Logo are trademarks of Red Hat, Inc.

The OpenStack word mark and the Square O Design, together or apart, are trademarks or registered trademarks of OpenStack Foundation in the United States and other countries, and are used with the OpenStack Foundation's permission.

Documentation updates

The title page of this document contains the follow ing identifying information:

- Softw are Version number, w hich indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Softw are Release Date, which indicates the release date of this version of the software.

To check for recent updates or to verify that you are using the most recent edition of a document, go to the follow ing URL and sign-in or register: https://softwaresupport.hpe.com.

Select Manuals from the Dashboard menu to view all available documentation. Use the search and filter functions to find documentation, w hitepapers, and other information sources.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your Hewlett Packard Enterprise sales representative for details.

Support

Visit the Hew lett Packard Enterprise Softw are Support Online web site at https://softwaresupport.hpe.com.