



HPE UCA Automation

UCA Autoconsole CA Guide for Linux
(RHEL 6.4)

Version: 2.1

Edition: 1.0

July 2016



Hewlett Packard
Enterprise

Notices

Legal notice

© Copyright 2016, Hewlett Packard Enterprise Development LP

Confidential computer software. Valid license from HPE 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.

The information contained herein is subject to change without notice. The only warranties for HPE products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HPE shall not be liable for technical or editorial errors or omissions contained herein.

Printed in the US

Trademarks

Adobe®, Acrobat® and PostScript® are trademarks of Adobe Systems Incorporated.

HP-UX Release 10.20 and later and HP-UX Release 11.00 and later (in both 32 and 64-bit configurations) on all HP 9000 computers are Open Group UNIX 95 branded products.

Oracle® and Java™ are registered trademark of Oracle and/or its affiliates.

Microsoft®, Internet Explorer, Windows®, Windows Server®, and Windows NT® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Firefox® is a registered trademark of the Mozilla Foundation.

Google Chrome® is a trademark of Google Inc.

Oracle® is a registered U.S. trademark of Oracle Corporation, Redwood City, California.

EnterpriseDB® is a registered trademark of EnterpriseDB.

Postgres Plus® Advanced Server is a registered U.S. trademark of EnterpriseDB.

UNIX® is a registered trademark of The Open Group.

X/Open® is a registered trademark, and the X device is a trademark of X/Open Company Ltd. in the UK and other countries.

Red Hat® is a registered trademark of the Red Hat Company.

Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

Neo4j is a trademark of Neo Technology.

Contents

Notices.....	1
Preface	6
About this guide.....	6
Audience.....	6
Software versions.....	6
Typographical conventions.....	6
Associated documents.....	6
Support	7
Chapter 1 Solution overview	8
1.1 Deliverables.....	9
Chapter 2 Installing UCA Autoconsole CA	10
Chapter 3 Configuring UCA-Autoconsole CA	11
Chapter 4 Uninstalling UCA Autoconsole CA.....	12
Chapter 5 UCA Autoconsole CA WSDL	14

List of tables

Table 1: Software versions..... 6

List of figures

Figure 1: Solution overview	8
Figure 2: Internal Structure.....	9

Preface

About this guide

This guide describes how to install, configure, and uninstall UCA Autoconsole CA.

Product Name: UCA Automation

Product Version: 2.1

Audience

This document is intended for the delivery team and the administrators who install and configure UCA Autoconsole CA.

Software versions

The term UNIX is used as a generic reference to the operating system, unless otherwise specified.

The software versions referred to in this document are as follows:

Table 1: Software versions

Product version	Supported operating systems
UCA Automation 2.1	Linux Red Hat Enterprise Linux Server release 6.4
OSS Open Mediation 7.2.0	Linux Red Hat Enterprise Linux Server release 6.4

Typographical conventions

Fixed width text	It is used for filenames and their contents, computer inputs or outputs, program codes, and so on.
<i>Italic text</i>	It is used for labels, parameters, emphasized text, and replaceable text, citations and references
Bold text	It is used to indicate navigation options in the interfaces; for example, the text appearing in buttons and menu items. User interface controls, window titles, generic emphasis
<angle brackets>	Indicates generic variable names that must be substituted by real values or strings.

Associated documents

The following documents contain useful reference information:

- OSS Open Mediation OSS Open Mediation V7.2.0 Linux Installation and Configuration Guide

Support

Please visit our HPE Software Support Online Web site at softwaresupport.hpe.com for contact information, and details about HPE Software products, services, and support.

The Software support area of the Software Web site includes the following:

- Downloadable documentation.
- Troubleshooting information.
- Patches and updates.
- Problem reporting.
- Training information.
- Support program information.

Chapter 1

Solution overview

The following is the high-level solution overview.

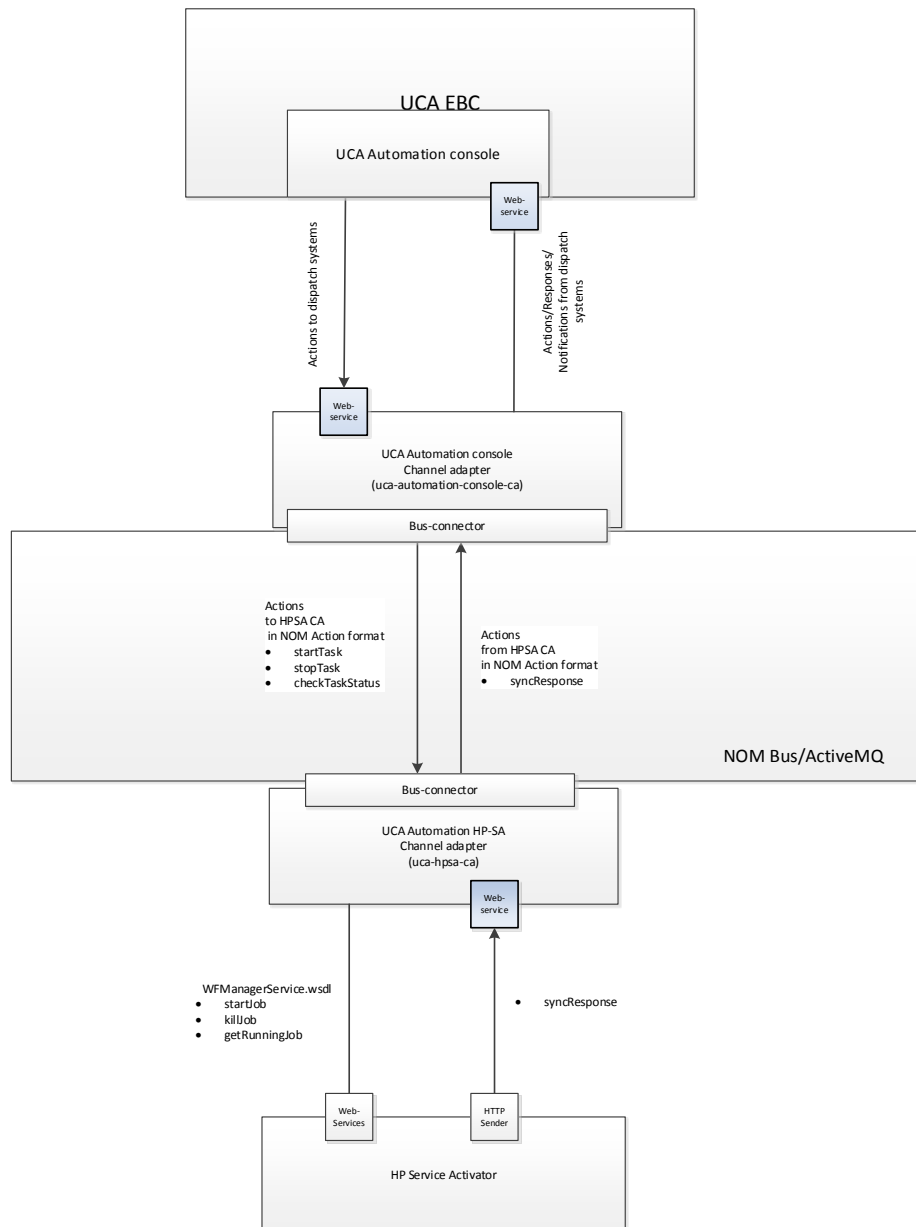


Figure 1: Solution overview

The following is the channel adapter internal structure. It consists from the number of components provided by NOM SDK to transform data, bind to specific Web-Services, and communicate with NOM Bus.

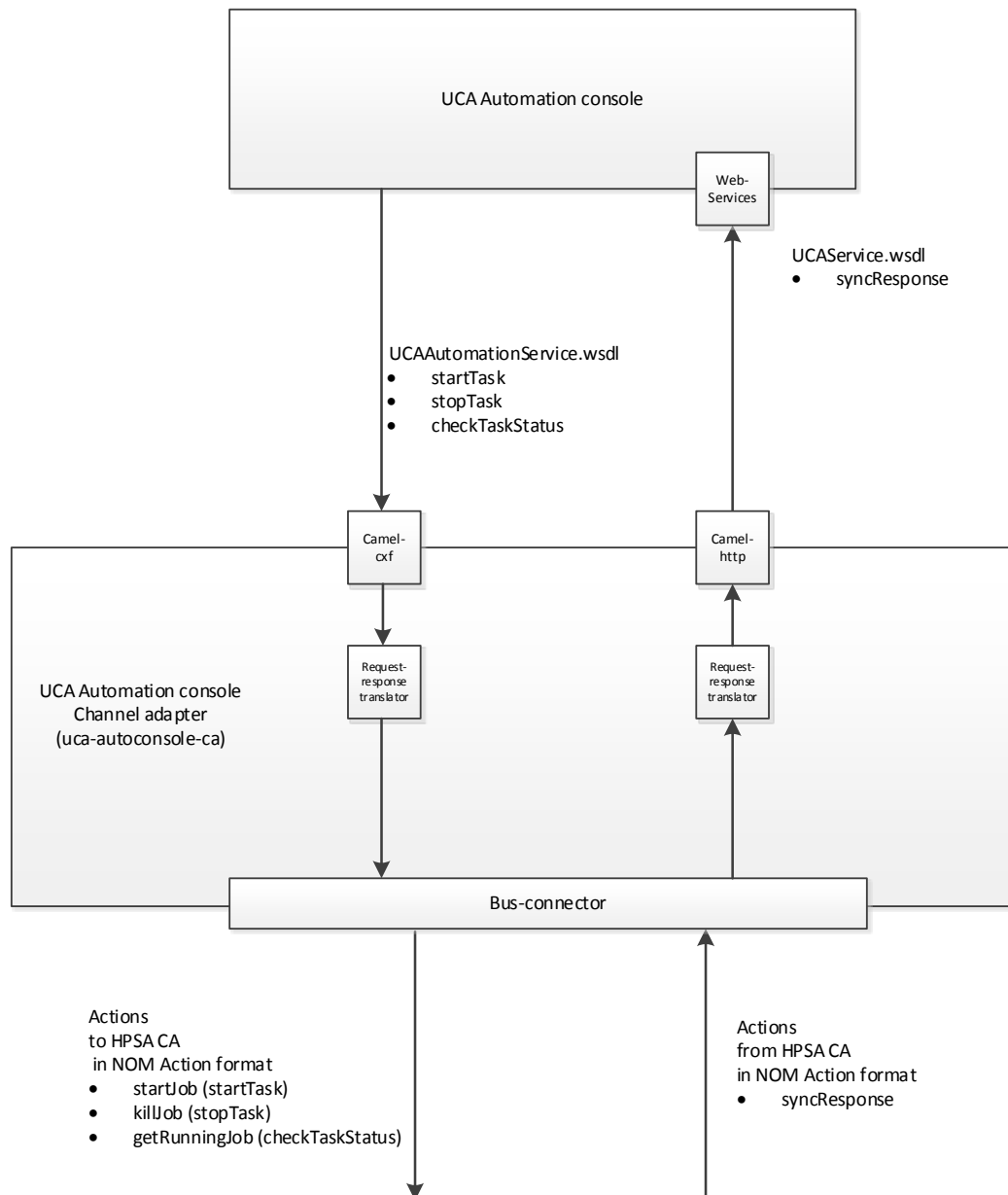


Figure 2: Channel Adapter structure

1.1 Deliverables

The `uca-autoconsole-ca-2.0.0-L.tar` contains a channel adaptor for integration with UCA console in scope of UCA Automation solution which is packed in RPM for specific operating system.

Chapter 2

Installing UCA Autoconsole CA

You can install the UCA Autoconsole CA manually using the following procedure.

Run the commands mentioned in this section as root user or a user who has the privileges to install RPM files and NOM IPs.

1. Extract the `uca-autoconsole-ca-2.0.0-L.tar` file using the following command:

```
tar xvf uca-autoconsole-ca-2.0.0-L.tar
```

2. Install the RPM to the `openmediation-72` directory using the following command.

```
rpm -i --relocate /opt/ngoss/=/opt/openmediation-72/ ngossuca-autoconsole-ca-2.0.0.noarch.rpm
```

3. Use the following commands to install UCA Autoconsole CA.

```
nom_admin --install-ip uca-autoconsole-ca-20  
nom_admin --install-ip-in-container uca-autoconsole-ca-20  
nom_admin --deploy-ip-in-container uca-autoconsole-ca-20
```

For some very specific needs UCA Autoconsole CA can be installed by a non-root user.

When installing UCA Automation as non-root user, the following limitations must be understood

The system RPM database is not accessible by a non-root user. As a consequence, when installation is performed by a non-root user, a specific RPM database must be specified. The default RPM repository for non-root installation is set to `~/rpmdb` (where `~` is the user home directory). A new RPM database can be initialized as follows

```
rpm --initdb --dbpath <alternate rpm db>
```

The new rpm db path can be specified when installing the CA

```
rpm -i --nodeps --dbpath <alternate rpm db> --relocate /opt/ngoss/=/app/NOM/opt/openmediation-72/ ngossuca-autoconsole-ca-2.0.0.noarch.rpm
```

Chapter 3

Configuring UCA-Autoconsole CA

Use the following procedure to configure UCA-Autoconsole CA.

1. Configure the `<installation package deployment directory>/etc/config.properties` property file with the following parameters:
 - a. **uca.uca-automation.host**—UCA console CA host for the exposed web-service.
 - b. **uca.uca-automation.port**—UCA console CA port for the exposed web-service.
 - c. **uca.console.service**—Name of the UCA console service.
 - d. **uca.console.host**—UCA Automation console host or IP address.
 - e. **uca.console.port**—UCA Automation console port number.
2. To apply the new configuration, redeploy the channel adaptor by using the `nom_admin` tool.

```
nom_admin --undeploy-ip-in-container uca-autoconsole-ca-20  
nom_admin --deploy-ip-in-container uca-autoconsole-ca-20
```

The default values are as follows:

```
# connectivity settings  
uca.uca-automation.host=localhost  
uca.uca-automation.port=12500  
uca.console.service=UCA_Automation_Foundation_UCA-V1.1-1A-UCAAutomation/UCAService  
uca.console.host=localhost  
uca.console.port=8888
```

Modify the value of `uca.console.service` as `UCA_Automation_Foundation_UCA-V2.1-1A-UCAAutomation/UCAService`

Chapter 4

Uninstalling UCA Autoconsole CA

Perform the following steps to manually uninstall UCA Autoconsole CA.

1. Undeploy the channel adaptor from any OSS Open Mediation container where it is deployed.

```
nom_admin --undeploy-ip-in-container uca-autoconsole-ca-20
```

2. Uninstall the channel adaptor from any OSS Open Mediation container.

```
nom_admin --remove-ip-in-container uca-autoconsole-ca-20
```

3. Uninstall the channel adapter from OSS Open Mediation.

```
nom_admin --remove-ip uca-autoconsole-ca-20
```

4. Erase the package from the system.

```
rpm -e ngossuca-autoconsole-ca-2.0.0-RHEL5.noarch
```

For a non-root user specify the alternate rpm database path

```
rpm -e --dbpath <alternate rpm db> ngossuca-autoconsole-ca-2.0.0-RHEL5.noarch
```


Chapter 5

UCA Autoconsole CA WSDL

The wsdl that defines the operations supported by the channel adapter is as follows

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://ws.ucaautomation.hp.com/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:ns1="http://types.ws.ucaautomation.hp.com/" xmlns="http://schemas.xmlsoap.org/wsdl/"
name="UCAAutomationConsoleService" targetNamespace="http://ws.ucaautomation.hp.com/">
  <types>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://ws.ucaautomation.hp.com/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:ns1="http://types.ws.ucaautomation.hp.com/" xmlns="http://schemas.xmlsoap.org/wsdl/"

  <xsd:import namespace="http://types.ws.ucaautomation.hp.com/"
schemaLocation="http://15.154.113.175:12500/UCAAutomationConsoleService/UCAAutomationConsol
eService?xsd=UCA.xsd"/>

</xsd:schema>
  </types>
  <message name="UCATaskException">
    <part element="ns1:UCATaskException" name="UCATaskException">
<documentation>Encapsulation of of all the Exceptions thrown when starting, stopping or checking
status of a task. This must encapsulate the Exception thrown and instance of
UCATaskExceptionBean</documentation>
    </part>
  </message>
  <message name="startTaskResponse">
    <part element="ns1:startTaskResponse" name="parameters">
<documentation>The task identifier. Represents the instance of the task on the Activation system. For
eg in HPSA this will contain the job id of the UCAController workflow</documentation>
    </part>
  </message>
  <message name="stopTask">
    <part element="ns1:stopTask" name="parameters">
<documentation>Contains details necessary to stop a Task.</documentation>
    </part>
  </message>
  <message name="checkTaskStatusResponse">
    <part element="ns1:checkTaskStatusResponse" name="parameters">
<documentation>The task status. Represents status of the task on the Activation system. We are only
intrested in whether the Task is running or not. For eg in HPSA this will contain the value of status
attribute of the RunningJobDescriptor returned when invoking getRunningJob</documentation>
    </part>
  </message>
```

```

<message name="startTask">
  <part element="ns1:startTask" name="parameters">
<documentation>Contains details necessary to start a Task.</documentation>
  </part>
</message>
<message name="checkTaskStatus">
  <part element="ns1:checkTaskStatus" name="parameters">
<documentation>Contains details necessary to check status of a Task.</documentation>
  </part>
</message>
<message name="stopTaskResponse">
  <part element="ns1:stopTaskResponse" name="parameters">
<documentation>Synchronous response sent back to UCA-Automation console to comply with the
InOut message exchange pattern. This indicates whether Task has been stopped. In case of HPSA
the stopJob has a void return. Hence the exception thrown could be used to decide the status of
stopTask</documentation>
  </part>
</message>
<portType name="UCAAutomationConsoleService">
  <operation name="startTask">
<documentation>starts a specified task on the Activation system. For eg startTask operation must
invoke startJob on HPSA</documentation>
    <input message="tns:startTask" name="startTask">
</input>
    <output message="tns:startTaskResponse" name="startTaskResponse">
</output>
    <fault message="tns:UCATaskException" name="UCATaskException">
</fault>
  </operation>
  <operation name="stopTask">
<documentation>stops a specified task on the Activation system. For eg stopTask operation must
invoke killJob on HPSA</documentation>
    <input message="tns:stopTask" name="stopTask">
</input>
    <output message="tns:stopTaskResponse" name="stopTaskResponse">
</output>
    <fault message="tns:UCATaskException" name="UCATaskException">
</fault>
  </operation>
  <operation name="checkTaskStatus">
<documentation>checks the status of a specified task on the Activation system. For eg
checkTaskStatus operation must invoke getRunningJob on HPSA</documentation>
    <input message="tns:checkTaskStatus" name="checkTaskStatus">
</input>
    <output message="tns:checkTaskStatusResponse" name="checkTaskStatusResponse">
</output>
    <fault message="tns:UCATaskException" name="UCATaskException">
</fault>
  </operation>
</portType>

```



```

<binding name="UCAAutomationConsoleServicePortBinding"
type="tns:UCAAutomationConsoleService">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="startTask">
    <soap:operation soapAction="" style="document"/>
    <input name="startTask">
      <soap:body use="literal"/>
    </input>
    <output name="startTaskResponse">
      <soap:body use="literal"/>
    </output>
    <fault name="UCATaskException">
      <soap:fault name="UCATaskException" use="literal"/>
    </fault>
  </operation>
  <operation name="stopTask">
    <soap:operation soapAction="" style="document"/>
    <input name="stopTask">
      <soap:body use="literal"/>
    </input>
    <output name="stopTaskResponse">
      <soap:body use="literal"/>
    </output>
    <fault name="UCATaskException">
      <soap:fault name="UCATaskException" use="literal"/>
    </fault>
  </operation>
  <operation name="checkTaskStatus">
    <soap:operation soapAction="" style="document"/>
    <input name="checkTaskStatus">
      <soap:body use="literal"/>
    </input>
    <output name="checkTaskStatusResponse">
      <soap:body use="literal"/>
    </output>
    <fault name="UCATaskException">
      <soap:fault name="UCATaskException" use="literal"/>
    </fault>
  </operation>
</binding>
<service name="UCAAutomationConsoleService">
  <port binding="tns:UCAAutomationConsoleServicePortBinding"
name="UCAAutomationConsoleServicePort">
    <soap:address
location="http://15.154.113.175:12500/UCAAutomationConsoleService/UCAAutomationConsoleService"
/>
  </port>
</service>
</definitions>

```