



HPE UCA Automation

HPE SA 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	5
About this guide.....	5
Audience.....	5
Software versions.....	5
Typographical conventions.....	5
Associated documents.....	6
Support	6
Chapter 1 Solution overview	7
1.1 Deliverables.....	8
Chapter 2 Installing UCA HPE SA CA	9
Chapter 3 Configuring UCA HPE SA CA	10
Chapter 4 Uninstalling UCA HPE SA CA	11
Chapter 5 SA CA WSDL	13

List of tables

Table 1: Software versions.....	5
---------------------------------	---

List of figures

Figure 1: Solution overview	7
Figure 2: Channel adaptor structure.....	8

Preface

About this guide

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

Product Name: UCA Automation

Product Version: 2.1

Read this document before installing or using this Software.

Audience

This document is intended for the delivery team and the administrators who install and configure UCA HPE SA 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 high-level solution overview. The UCA HPE SA CA acts as a bridge between the HPE Service Activator and other components integrated through NOM Bus. It transforms actions from and to the UCA Automated Console system HPE SA format and back.

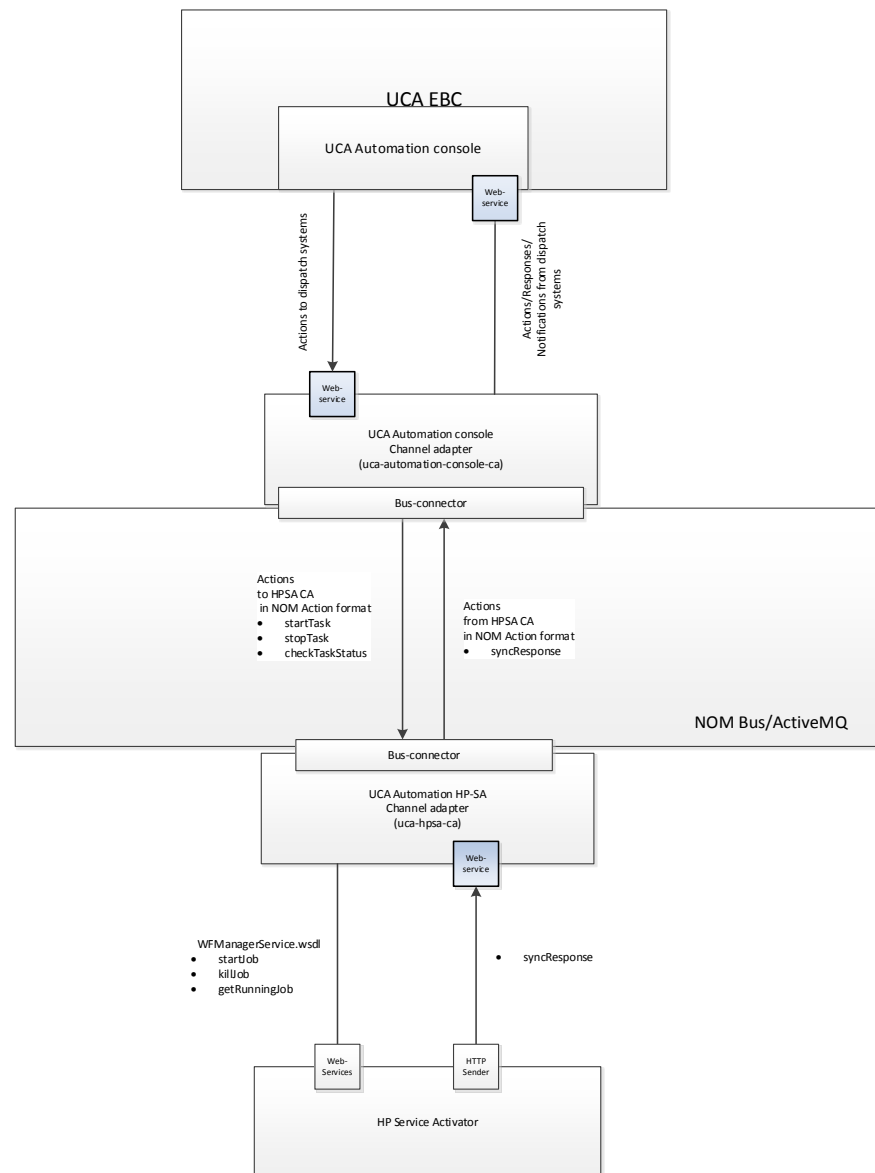


Figure 1: Solution overview

The following is the channel adaptor internal structure.

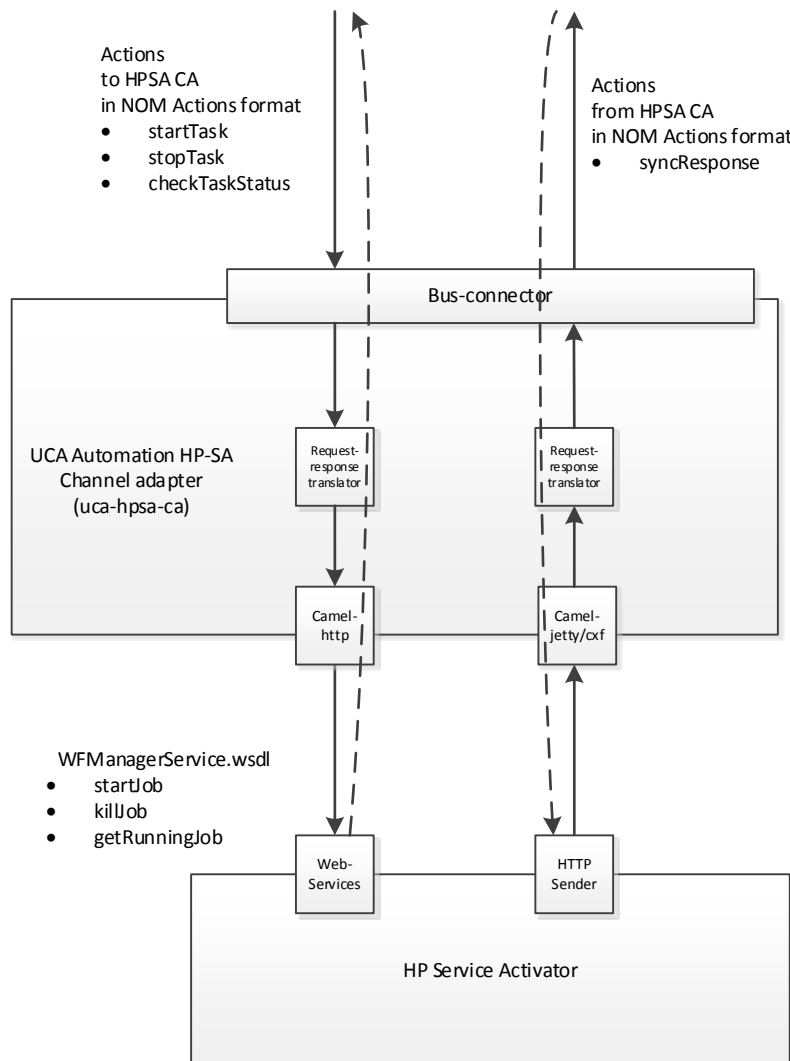


Figure 2: Channel adaptor structure

1.1 Deliverables

The `uca-hpsa-ca-2.0.0-L.tar` contains a channel adaptor for integration with HPE SA in scope of UCA Automation solution.

Chapter 2

Installing UCA HPE SA CA

You can install UCA HPE SA 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-hpsa-ca-2.0.0-L.tar` file using the following command:

```
tar xvf uca-hpsa-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-hpsa-ca-2.0.0.x86_64.rpm
```

3. Use the following commands to install UCA HPE SA CA.

```
nom_admin --install-ip uca-hpsa-ca-20  
nom_admin --install-ip-in-container uca-hpsa-ca-20  
nom_admin --deploy-ip-in-container uca-hpsa-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=/opt/openmediation-72/  
ngossuca-hpsa-ca-2.0.0.x86_64.rpm
```

Chapter 3

Configuring UCA HPE SA CA

Use the following procedure to configure UCA HPE SA CA.

1. Configure the <installation package deployment directory>/etc/config.properties property file with the following parameters:
 - a. **hpsa.host**—HPE SA hostname/IP address
 - b. **hpsa.port**—HPE SA host port number
 - c. **hpsa.userid**—HPE SA username
 - d. **hpsa.password**—HPE SA user password
 - e. **hpsa.controller.workflow.name**—UCA Automation controller workflow name
 - f. **hpsa.uca-automation.sync-service.host**—Hostname/IP address that is exposed by the UCA HPE SA CA Web service receives the requests from HPE SA (sender module)
 - g. **hpsa.uca-automation.sync-service.port**—Port number that is exposed by the UCA HPE SA CA Web service receives the requests from HPE SA (sender module)
2. To apply the new configuration, redeploy the channel adaptor by using the nom_admin tool.

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

The default values are as follows:

```
# HPSA connectivity settings
hpsa.host=localhost
hpsa.port=8080
hpsa.userid=sa
hpsa.password=sa

# UCA Automation controller workflow
hpsa.controller.workflow.name=UCAController

# UCA Automation response handler connectivity settings
hpsa.uca-automation.sync-service.host=localhost
hpsa.uca-automation.sync-service.port=8191
```

Chapter 4

Uninstalling UCA HPE SA CA

Perform the following steps to uninstall UCA HPE SA CA manually.

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

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

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

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

3. Uninstall the channel adapter from OSS Open Mediation.

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

4. Erase the package from the system.

```
rpm -e ngossuca-hpsa-ca-2.0.0-RHEL5.x86_64
```

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

```
rpm -e --dbpath <alternate rpm db> ngossuca-hpsa-ca-2.0.0-RHEL5.x86_64
```


Chapter 5

SA CA WSDL

The wsdl definition for HPE SA

Operation	Description
startJob	<p>Starts up the given workflow and stores in the database the given data for a further use inside the running workflow if the user is authenticated properly.</p> <p>The user will only be allowed to start a new job if he has the proper roles according to the permissions set in the workflow definition (see the roles defined in the <Default-Role> and <Start-Role> tags in the workflow definition).</p> <p>Parameters</p> <p>userId : The user allowed to start a workflow</p> <p>password: The user password</p> <p>workflowName : The name of the workflow to be started</p> <p>data: additional data to store in the database before the workflow is started up or an empty String (not null) if this is not required by the workflow.</p>
killJob	<p>Kills the specified job if the user is successfully authenticated. This method returns immediately, it won't wait for the job to terminate.</p> <p>The user will only be allowed to kill a job if he has the proper roles according to the permissions set in the workflow definition (see the roles defined in the <Kill-Role> tag in the workflow definition). If there are no permission roles set in the workflow definition then every operation over this workflow is allowed for every user. Also, if no authentication module is configured then every job will be accessible for every user</p> <p>Parameters</p> <p>userId - the name of the user who invokes the method.</p> <p>password - the password of the user who invokes the method.</p> <p>jobId - the identifier of the job to be killed</p>
getRunningJob	<p>Gets the information of a running job identified by its job identifier independently of the cluster nodes where it may be running.</p> <p>The user will only be allowed to get a running job if he has the proper roles according to the permissions set in the workflow definition (see the roles defined in the <Trace-Role> tag in the workflow definition). If there are no permission roles set in the workflow definition then every operation over this workflow is allowed for every user. Also, if no authentication module is configured then every job will be accessible for every user.</p> <p>Parameters</p> <p>userId - the name of the user who invokes the method.</p> <p>password - the password of the user who invokes the method.</p> <p>jobId - the identifier of the job to retrieve.</p>

```

<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions name="WFManagerWSService" targetNamespace="http://ws.activator.ov.hp.com/"
xmlns:ns1="http://jaxb.dev.java.net/array" xmlns:ns2="http://schemas.xmlsoap.org/soap/http"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://ws.activator.ov.hp.com/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <wsdl:types>
    <xs:schema attributeFormDefault="unqualified" elementFormDefault="unqualified"
targetNamespace="http://ws.activator.ov.hp.com/" xmlns:tns="http://ws.activator.ov.hp.com/"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
      <xs:complexType name="hashMapWrap">
        <xs:sequence>
          <xs:element name="map">
            <xs:complexType>
              <xs:sequence>
                <xs:element maxOccurs="unbounded" minOccurs="0" name="entry">
                  <xs:complexType>
                    <xs:sequence>
                      <xs:element minOccurs="0" name="key" type="xs:string"/>
                      <xs:element minOccurs="0" name="value" type="xs:string"/>
                    </xs:sequence>
                  </xs:complexType>
                </xs:element>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="RunningJobDescriptor">
        <xs:sequence>
          <xs:element name="jobId" nillable="true" type="xs:long"/>
          <xs:element minOccurs="0" name="name" nillable="true" type="xs:string"/>
          <xs:element minOccurs="0" name="description" nillable="true" type="xs:string"/>
          <xs:element minOccurs="0" name="status" nillable="true" type="xs:string"/>
          <xs:element minOccurs="0" name="stepName" nillable="true" type="xs:string"/>
          <xs:element minOccurs="0" name="startTime" nillable="true" type="xs:dateTime"/>
          <xs:element minOccurs="0" name="hostName" nillable="true" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
      <xs:element name="MalformedURLException" type="tns:MalformedURLException"/>
      <xs:complexType name="MalformedURLException">
        <xs:sequence/>
      </xs:complexType>
      <xs:element name="AuthException" type="tns:AuthException"/>
      <xs:complexType name="AuthException">
        <xs:sequence/>
      </xs:complexType>
      <xs:element name="WFException" type="tns:WFException"/>
      <xs:complexType name="WFException">

```

```

<xs:sequence/>
</xs:complexType>
<xs:element name="IOException" type="tns:IOException"/>
<xs:complexType name="IOException">
  <xs:sequence/>
</xs:complexType>
<xs:element name="NotBoundException" type="tns:NotBoundException"/>
<xs:complexType name="NotBoundException">
  <xs:sequence/>
</xs:complexType>
<xs:element name="WFNotAuthorizedException" type="tns:WFNotAuthorizedException"/>
<xs:complexType name="WFNotAuthorizedException">
  <xs:sequence/>
</xs:complexType>
<xs:element name="SyncFailedException" type="tns:SyncFailedException"/>
<xs:complexType name="SyncFailedException">
  <xs:sequence/>
</xs:complexType>
<xs:element name="FileNotFoundException" type="tns:FileNotFoundException"/>
<xs:complexType name="FileNotFoundException">
  <xs:sequence/>
</xs:complexType>
<xs:element name="WFConfigException" type="tns:WFConfigException"/>
<xs:complexType name="WFConfigException">
  <xs:sequence/>
</xs:complexType>
<xs:element name="WFDBException" type="tns:WFDBException"/>
<xs:complexType name="WFDBException">
  <xs:sequence/>
</xs:complexType>
<xs:element name="WFNoSuchJobException" type="tns:WFNoSuchJobException"/>
<xs:complexType name="WFNoSuchJobException">
  <xs:sequence/>
</xs:complexType>
<xs:element name="WFSuspendedException" type="tns:WFSuspendedException"/>
<xs:complexType name="WFSuspendedException">
  <xs:sequence/>
</xs:complexType>
<xs:element name="WFConnectivityException" type="tns:WFConnectivityException"/>
<xs:complexType name="WFConnectivityException">
  <xs:sequence>
    <xs:element name="permanent" nillable="true" type="xs:boolean"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="WFAlreadyKilledException" type="tns:WFAlreadyKilledException"/>
<xs:complexType name="WFAlreadyKilledException">
  <xs:sequence/>
</xs:complexType>
...

```



```

</xs:schema>
<xs:schema targetNamespace="http://jaxb.dev.java.net/array" version="1.0"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:complexType final="#all" name="stringArray">
  <xs:sequence>
    <xs:element maxOccurs="unbounded" minOccurs="0" name="item" nillable="true"
type="xs:string"/>
  </xs:sequence>
</xs:complexType>
</xs:schema>
</wsdl:types>
...
...
<wsdl:message name="startJob">
  <wsdl:part name="userId" type="xsd:string">
</wsdl:part>
  <wsdl:part name="password" type="xsd:string">
</wsdl:part>
  <wsdl:part name="workflowname" type="xsd:string">
</wsdl:part>
  <wsdl:part name="data" type="xsd:string">
</wsdl:part>
</wsdl:message>
<wsdl:message name="killJob">
  <wsdl:part name="userId" type="xsd:string">
</wsdl:part>
  <wsdl:part name="password" type="xsd:string">
</wsdl:part>
  <wsdl:part name="jobId" type="xsd:long">
</wsdl:part>
</wsdl:message>
<wsdl:message name="getRunningJob">
  <wsdl:part name="userId" type="xsd:string">
</wsdl:part>
  <wsdl:part name="password" type="xsd:string">
</wsdl:part>
  <wsdl:part name="jobId" type="xsd:long">
</wsdl:part>
</wsdl:message>
<wsdl:message name="startJobResponse">
  <wsdl:part name="uniqueJobIdentifier" type="xsd:long">
</wsdl:part>
</wsdl:message>
<wsdl:message name="killJobResponse">
</wsdl:message>
<wsdl:message name="getRunningJobResponse">
  <wsdl:part name="runningJob" type="tns:RunningJobDescriptor">
</wsdl:part>
</wsdl:message>
...

```

```
...
<wsdl:portType name="WFManager">
  <wsdl:operation name="startJob">
    <wsdl:input message="tns:startJob" name="startJob">
  </wsdl:input>
    <wsdl:output message="tns:startJobResponse" name="startJobResponse">
  </wsdl:output>
    <wsdl:fault message="tns:MalformedURLException" name="MalformedURLException">
  </wsdl:fault>
    <wsdl:fault message="tns:AuthException" name="AuthException">
  </wsdl:fault>
    <wsdl:fault message="tns:WFException" name="WFException">
  </wsdl:fault>
    <wsdl:fault message="tns:IOException" name="IOException">
  </wsdl:fault>
    <wsdl:fault message="tns:NotBoundException" name="NotBoundException">
  </wsdl:fault>
    <wsdl:fault message="tns:WFNotAuthorizedException" name="WFNotAuthorizedException">
  </wsdl:fault>
    <wsdl:fault message="tns:SyncFailedException" name="SyncFailedException">
  </wsdl:fault>
    <wsdl:fault message="tns:FileNotFoundException" name="FileNotFoundException">
  </wsdl:fault>
    <wsdl:fault message="tns:WFConfigException" name="WFConfigException">
  </wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="killJob">
    <wsdl:input message="tns:killJob" name="killJob">
  </wsdl:input>
    <wsdl:output message="tns:killJobResponse" name="killJobResponse">
  </wsdl:output>
    <wsdl:fault message="tns:MalformedURLException" name="MalformedURLException">
  </wsdl:fault>
    <wsdl:fault message="tns:WFDBException" name="WFDBException">
  </wsdl:fault>
    <wsdl:fault message="tns:AuthException" name="AuthException">
  </wsdl:fault>
    <wsdl:fault message="tns:WFNoSuchJobException" name="WFNoSuchJobException">
  </wsdl:fault>
    <wsdl:fault message="tns:NotBoundException" name="NotBoundException">
  </wsdl:fault>
    <wsdl:fault message="tns:WFNotAuthorizedException" name="WFNotAuthorizedException">
  </wsdl:fault>
    <wsdl:fault message="tns:WFSuspendedException" name="WFSuspendedException">
  </wsdl:fault>
    <wsdl:fault message="tns:WFConnectivityException" name="WFConnectivityException">
  </wsdl:fault>
    <wsdl:fault message="tns:WFAlreadyKilledException" name="WFAlreadyKilledException">
  </wsdl:fault>
  </wsdl:operation>
</wsdl:portType>

```

```

<wsdl:operation name="getRunningJob">
  <wsdl:input message="tns:getRunningJob" name="getRunningJob">
</wsdl:input>
  <wsdl:output message="tns:getRunningJobResponse" name="getRunningJobResponse">
</wsdl:output>
  <wsdl:fault message="tns:MalformedURLException" name="MalformedURLException">
</wsdl:fault>
  <wsdl:fault message="tns:WFDBException" name="WFDBException">
</wsdl:fault>
  <wsdl:fault message="tns:AuthException" name="AuthException">
</wsdl:fault>
  <wsdl:fault message="tns:WFNoSuchJobException" name="WFNoSuchJobException">
</wsdl:fault>
  <wsdl:fault message="tns:NotBoundException" name="NotBoundException">
</wsdl:fault>
  <wsdl:fault message="tns:WFNotAuthorizedException" name="WFNotAuthorizedException">
</wsdl:fault>
  <wsdl:fault message="tns:WFConnectivityException" name="WFConnectivityException">
</wsdl:fault>
</wsdl:operation>
...
...
</wsdl:portType>
<wsdl:binding name="WFManagerWSServiceSoapBinding" type="tns:WFManager">
  <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="startJob">
    <soap:operation soapAction="" style="rpc"/>
    <wsdl:input name="startJob">
      <soap:body namespace="http://ws.activator.ov.hp.com/" use="literal"/>
    </wsdl:input>
    <wsdl:output name="startJobResponse">
      <soap:body namespace="http://ws.activator.ov.hp.com/" use="literal"/>
    </wsdl:output>
    <wsdl:fault name="MalformedURLException">
      <soap:fault name="MalformedURLException" use="literal"/>
    </wsdl:fault>
    <wsdl:fault name="AuthException">
      <soap:fault name="AuthException" use="literal"/>
    </wsdl:fault>
    <wsdl:fault name="WFException">
      <soap:fault name="WFException" use="literal"/>
    </wsdl:fault>
    <wsdl:fault name="IOException">
      <soap:fault name="IOException" use="literal"/>
    </wsdl:fault>
    <wsdl:fault name="NotBoundException">
      <soap:fault name="NotBoundException" use="literal"/>
    </wsdl:fault>
    <wsdl:fault name="WFNotAuthorizedException">
      <soap:fault name="WFNotAuthorizedException" use="literal"/>

```

```

</wsdl:fault>
<wsdl:fault name="SyncFailedException">
  <soap:fault name="SyncFailedException" use="literal"/>
</wsdl:fault>
<wsdl:fault name="FileNotFoundException">
  <soap:fault name="FileNotFoundException" use="literal"/>
</wsdl:fault>
<wsdl:fault name="WFConfigException">
  <soap:fault name="WFConfigException" use="literal"/>
</wsdl:fault>
</wsdl:operation>
<wsdl:operation name="killJob">
  <soap:operation soapAction="" style="rpc"/>
  <wsdl:input name="killJob">
    <soap:body namespace="http://ws.activator.ov.hp.com/" use="literal"/>
  </wsdl:input>
  <wsdl:output name="killJobResponse">
    <soap:body namespace="http://ws.activator.ov.hp.com/" use="literal"/>
  </wsdl:output>
  <wsdl:fault name="MalformedURLException">
    <soap:fault name="MalformedURLException" use="literal"/>
  </wsdl:fault>
  <wsdl:fault name="WFDBException">
    <soap:fault name="WFDBException" use="literal"/>
  </wsdl:fault>
  <wsdl:fault name="AuthException">
    <soap:fault name="AuthException" use="literal"/>
  </wsdl:fault>
  <wsdl:fault name="WFNoSuchJobException">
    <soap:fault name="WFNoSuchJobException" use="literal"/>
  </wsdl:fault>
  <wsdl:fault name="NotBoundException">
    <soap:fault name="NotBoundException" use="literal"/>
  </wsdl:fault>
  <wsdl:fault name="WFNotAuthorizedException">
    <soap:fault name="WFNotAuthorizedException" use="literal"/>
  </wsdl:fault>
  <wsdl:fault name="WFSuspendedException">
    <soap:fault name="WFSuspendedException" use="literal"/>
  </wsdl:fault>
  <wsdl:fault name="WFConnectivityException">
    <soap:fault name="WFConnectivityException" use="literal"/>
  </wsdl:fault>
  <wsdl:fault name="WFAlreadyKilledException">
    <soap:fault name="WFAlreadyKilledException" use="literal"/>
  </wsdl:fault>
</wsdl:operation>
<wsdl:operation name="getRunningJob">
  <soap:operation soapAction="" style="rpc"/>
  <wsdl:input name="getRunningJob">

```

```

    <soap:body namespace="http://ws.activator.ov.hp.com/" use="literal"/>
  </wsdl:input>
  <wsdl:output name="getRunningJobResponse">
    <soap:body namespace="http://ws.activator.ov.hp.com/" use="literal"/>
  </wsdl:output>
  <wsdl:fault name="MalformedURLException">
    <soap:fault name="MalformedURLException" use="literal"/>
  </wsdl:fault>
  <wsdl:fault name="WFDBException">
    <soap:fault name="WFDBException" use="literal"/>
  </wsdl:fault>
  <wsdl:fault name="AuthException">
    <soap:fault name="AuthException" use="literal"/>
  </wsdl:fault>
  <wsdl:fault name="WFNoSuchJobException">
    <soap:fault name="WFNoSuchJobException" use="literal"/>
  </wsdl:fault>
  <wsdl:fault name="NotBoundException">
    <soap:fault name="NotBoundException" use="literal"/>
  </wsdl:fault>
  <wsdl:fault name="WFNotAuthorizedException">
    <soap:fault name="WFNotAuthorizedException" use="literal"/>
  </wsdl:fault>
  <wsdl:fault name="WFConnectivityException">
    <soap:fault name="WFConnectivityException" use="literal"/>
  </wsdl:fault>
</wsdl:operation>
...
...
</wsdl:binding>
<wsdl:service name="WFManagerWSService">
  <wsdl:port binding="tns:WFManagerWSServiceSoapBinding" name="WFManagerPort">
    <soap:address location="http://localhost:8080/mwfmsoap/WFManagerService"/>
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

The wsdl definition for UCA Automation

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- Published by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is JAX-WS RI 2.2.8 svn-
revision#13980. -->
<!-- Generated by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is JAX-WS RI 2.2.8 svn-
revision#13980. -->
<definitions xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd" xmlns:wsp="http://www.w3.org/ns/ws-policy"
xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"

```

```

xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://ws.ucaautomation.hp.com/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/"
name="UCAServiceImplService" targetNamespace="http://ws.ucaautomation.hp.com/"
  <types>
    <xsd:schema>
      <xsd:import namespace="http://types.ws.ucaautomation.hp.com/"
schemaLocation="http://15.154.113.175:8888/UCA_Automation_Foundation_UCA-V2.1-1A-
UCAAutomation/UCAService?xsd=1"/>
    </xsd:schema>
  </types>
  <message name="terminate">
    <part name="parameters" element="ns1:terminate"
xmlns:ns1="http://types.ws.ucaautomation.hp.com/"/>
  </message>
  <message name="terminateResponse">
    <part name="parameters" element="ns2:terminateResponse"
xmlns:ns2="http://types.ws.ucaautomation.hp.com/"/>
  </message>
  <message name="UCAServiceException">
    <part name="fault" element="ns3:UCAServiceException"
xmlns:ns3="http://types.ws.ucaautomation.hp.com/"/>
  </message>
  <message name="synchResponse">
    <part name="parameters" element="ns4:synchResponse"
xmlns:ns4="http://types.ws.ucaautomation.hp.com/"/>
  </message>
  <message name="synchResponseResponse">
    <part name="parameters" element="ns5:synchResponseResponse"
xmlns:ns5="http://types.ws.ucaautomation.hp.com/"/>
  </message>
  <message name="initiate">
    <part name="parameters" element="ns6:initiate"
xmlns:ns6="http://types.ws.ucaautomation.hp.com/"/>
  </message>
  <message name="initiateResponse">
    <part name="parameters" element="ns7:initiateResponse"
xmlns:ns7="http://types.ws.ucaautomation.hp.com/"/>
  </message>
  <portType name="UCAService">
    <operation name="terminate">
      <input message="tns:terminate"
wsam:Action="http://ws.ucaautomation.hp.com/UCAService/terminateRequest"/>
      <output message="tns:terminateResponse"
wsam:Action="http://ws.ucaautomation.hp.com/UCAService/terminateResponse"/>
      <fault name="UCAServiceException" message="tns:UCAServiceException"
wsam:Action="http://ws.ucaautomation.hp.com/UCAService/terminate/Fault/UCAServiceException"/>
    </operation>
    <operation name="synchResponse">

```

```

        <input message="tns:synchResponse"
wsam:Action="http://ws.ucaautomation.hp.com/UCAService/synchResponseRequest"/>
        <output message="tns:synchResponseResponse"
wsam:Action="http://ws.ucaautomation.hp.com/UCAService/synchResponseResponse"/>
        <fault name="UCAServiceException" message="tns:UCAServiceException"
wsam:Action="http://ws.ucaautomation.hp.com/UCAService/synchResponse/Fault/UCAServiceExcept
ion"/>
    </operation>
    <operation name="initiate">
        <input message="tns:initiate"
wsam:Action="http://ws.ucaautomation.hp.com/UCAService/initiateRequest"/>
        <output message="tns:initiateResponse"
wsam:Action="http://ws.ucaautomation.hp.com/UCAService/initiateResponse"/>
        <fault name="UCAServiceException" message="tns:UCAServiceException"
wsam:Action="http://ws.ucaautomation.hp.com/UCAService/initiate/Fault/UCAServiceException"/>
    </operation>
</portType>
<binding name="UCAServicePortBinding" type="tns:UCAService">
    <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="terminate">
        <soap:operation soapAction=""/>
        <input>
            <soap:body use="literal"/>
        </input>
        <output>
            <soap:body use="literal"/>
        </output>
        <fault name="UCAServiceException">
            <soap:fault name="UCAServiceException" use="literal"/>
        </fault>
    </operation>
    <operation name="synchResponse">
        <soap:operation soapAction=""/>
        <input>
            <soap:body use="literal"/>
        </input>
        <output>
            <soap:body use="literal"/>
        </output>
        <fault name="UCAServiceException">
            <soap:fault name="UCAServiceException" use="literal"/>
        </fault>
    </operation>
    <operation name="initiate">
        <soap:operation soapAction=""/>
        <input>
            <soap:body use="literal"/>
        </input>
        <output>

```

```
        <soap:body use="literal"/>
    </output>
    <fault name="UCAServiceException">
        <soap:fault name="UCAServiceException" use="literal"/>
    </fault>
</operation>
</binding>
<service name="UCAServicImplService">
    <port name="UCAServicePort" binding="tns:UCAServicePortBinding">
        <soap:address
location="http://15.154.113.175:8888/UCA_Automation_Foundation_UCA-V2.1-1A-
UCAAutomation/UCAService"/>
    </port>
</service>
</definitions>
```