
HP UCA Automation



Version 1.1

**Installation Guide
for Linux (RHEL 6.4)**

Edition: 1.0

October 2014

© Copyright 2014 Hewlett-Packard Development Company, L.P.

Legal Notices

Warranty

The information contained herein is subject to change without notice. The only warranties for HP 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. HP shall not be liable for technical or editorial errors or omissions contained herein.

License Requirement and U.S. Government Legend

Confidential computer software. Valid license from HP 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 Notices

© Copyright 2014 Hewlett-Packard Development Company, L.P.

Trademark Notices

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.

Java™ is a 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 a registered trademark of Linus Torvalds in the U.S. and other countries.

Neo4j is a trademark of Neo Technology.

Contents

Legal Notices	2
Contents	3
Tables	5
Preface	6
Intended audience	6
Software versions	6
Typographical conventions.....	6
Associated documents.....	6
Support	7
Chapter 1	8
Introduction.....	8
1.1 Local install descriptors	8
Chapter 2	9
System requirements	9
2.1 Server platforms	9
2.2 Hardware requirements	9
2.3 Software requirements	9
2.3.1 HP UCA-EBC	9
2.3.1.1 Configure HP UCA EBC	9
2.3.2 HP Service Activator	10
2.3.3 NOM	11
2.3.4 Configure TeMIP Channel Adaptor for UCA Automation	12
2.3.5 JAVA.....	16
2.3.6 Configure TeMIP 6.2L	16
2.3.7 Configure TeMIP when used as NMS	16
2.4 Web client	17
Chapter 3	18
UCA Automation solution pack	18
3.1 Installing UCA Automation solution	18
3.2 Installing HPSA Foundation value pack.....	19
3.2.1 Deploy HPSA Foundation value pack	19
3.2.2 Configure HPSA Foundation value pack.....	20
3.3 Installing UCA EBC Foundation value pack	21
3.3.1 Configure UCA Automation UI	21
3.4 Installing UCA Automation UI	22
3.4.1 Deploy UCA Automation UI.....	22
Chapter 4	23
UCA Automation Orchestrator	23

4.1	Prerequisites for installing Orchestrator	23
4.2	Install UCA Automation Orchestrator	23
4.2.1	Install EMF plug-in.....	23
4.2.2	Install GEF plug-in	24
4.2.3	Install Windows builder	24
4.3	Configure Orchestrator.....	25
4.4	Decision tree command line utility.....	26
Chapter 5		28
Install NOM channel adaptors.....		28
5.1	HPSA channel adaptor	28
5.2	UCA Automation console channel adaptor	29
Chapter 6		31
UCA Automation licensing		31
6.1	Get a UCA Automation license	31
6.2	License policy	31
6.3	Use Webware website for product licenses	32
6.4	Install license keys for UCA Automation	33
6.5	Remove license keys for UCA Automation	33
Chapter 7		34
Code signing.....		34
7.1	Install and configure Gnu Privacy Guard (GnuGP)	34
7.2	Verify authenticity and integrity in RHEL 6.4	34
Glossary		35

Tables

Table 1 Software versions	6
Table 2 Local install descriptors.....	8
Table 3 RPM artifacts.....	19
Table 4 Neo4j configuration descriptors.....	27
Table 5 HPSA CA config descriptors.....	29
Table 6 UCA Console CA config descriptors	30

Preface

This guide describes how to install the product on the supported platform.

Product Name: UCA Automation

Product Version: 1.1

Read this document before installing or using this software.

Intended audience

This document is intended for the solution developers and software development engineers.

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.

Product version	Supported operating systems
UCA Automation 1.1	Linux Red Hat Enterprise Linux Server release 6.4

Table 1 Software versions

Typographical conventions

<code>Courier Text</code>	It is used for filenames and their contents, computer inputs or outputs, program codes, and so on.
<i>Italicized text</i>	It is used for labels, parameters, emphasized text, and replaceable text.
bold text	It is used to indicate navigation options in the interfaces; for example, the text appearing in buttons and menu items.
<angle brackets>	Indicates generic variable names that must be substituted by real values or strings.

Associated documents

The following documents contain useful reference information:

- HP UCA for Event Based Correlation Installation Guide
- HP UCA for Event Based Correlation Topology Extension Guide
- HP UCA for Event Based Correlation Value Pack Development Guide
- Deployment Manager (HPSA) Guide

- NOM Installation and Configuration Guide
- UCA HPSA CA Main Release Guide
- UCA Autoconsole CA Main Release Guide
- NOM UCA EBC Channel Adapter Installation Guide
- NOM TEMIP Channel Adapter Installation Guide

Support

Visit the HP Software Support Online Web site at www.hp.com/go/hpsoftwaresupport for contact information, and details about HP 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.

Introduction

This guide describes the installation procedure for the UCA Automation solution.

1.1 Local install descriptors

The following locations are used to define install locations throughout this guide.

Descriptor	Description
<code>\${ACTIVATOR_OPT}</code>	The base install directory of Service Activator. The UNIX® location is <code>/opt/OV/ServiceActivator</code> .
<code>\${SOLUTION_ETC}</code>	The <code>etc</code> directory of the HPSA value pack solution.
<code>\${UCA_EBC_HOME}</code>	The root directory of UCA-EBC. The default value is <code>/opt/UCA-EBC</code> .
<code>\${UCA_EBC_DATA}</code>	The data directory of UCA-EBC. The default value is <code>/var/opt/UCA-EBC</code> .
<code>\${UCA_EBC_INSTANCES}</code>	This directory might contain multiple instances of UCA-EBC where the value packs are deployed. The path refers to <code>\${UCA_EBC_DATA}/instances/default</code> .
<code>\${NOM_INSTANCE}</code>	<code>/var/opt/openmediation-70/containers/<instance-#></code>

Table 2 Local install descriptors

System requirements

2.1 Server platforms

Red Hat Enterprise Linux 6.4

- UCA-EBC 3.1
- HPSA V6.2-1A
- NOM 7.0

2.2 Hardware requirements

- X86-64 based system
- At least 4 GB of memory
- The database system requires a space for an Oracle 11g or an Enterprise Database Postgres 9.2 database instance of at least 4 GB for the product data.

2.3 Software requirements

2.3.1 HP UCA-EBC

- UCA for Event Based Correlation Server V3.1 and the latest patches.
Installing the UCA-EBC 3.1 server patch 00002 is mandatory.
- UCA for Event Based Correlation Topology Extension.

2.3.1.1 Configure HP UCA EBC

1. Edit the `${UCA_EBC_INSTANCES}/conf/uca-ebc.properties` file to add the following configuration to integrate the UCA Automation UI login with UCA EBC.

```
UCA Automation Foundation UCA-V1.1-1A-UCAAutomation-webapp-parameters=username=${user},userrole=${role}
```

2. Update the `${UCA_EBC_INSTANCES}/conf/GraphDisplayProfiles.xml` file with the following configuration.

```
<Profile name="ucaatm" displayName="Decision Tree View"
defaultProfile="true">
  <DefaultNode>
    <Icon>
      <MainIcon>images/round.jpg</MainIcon>
```

```

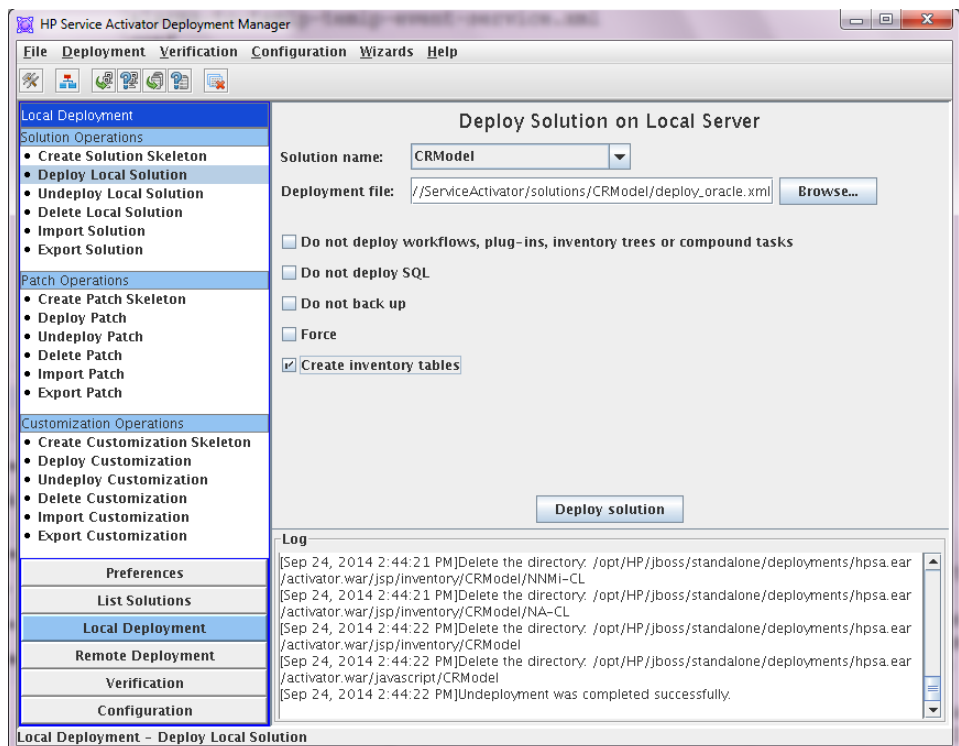
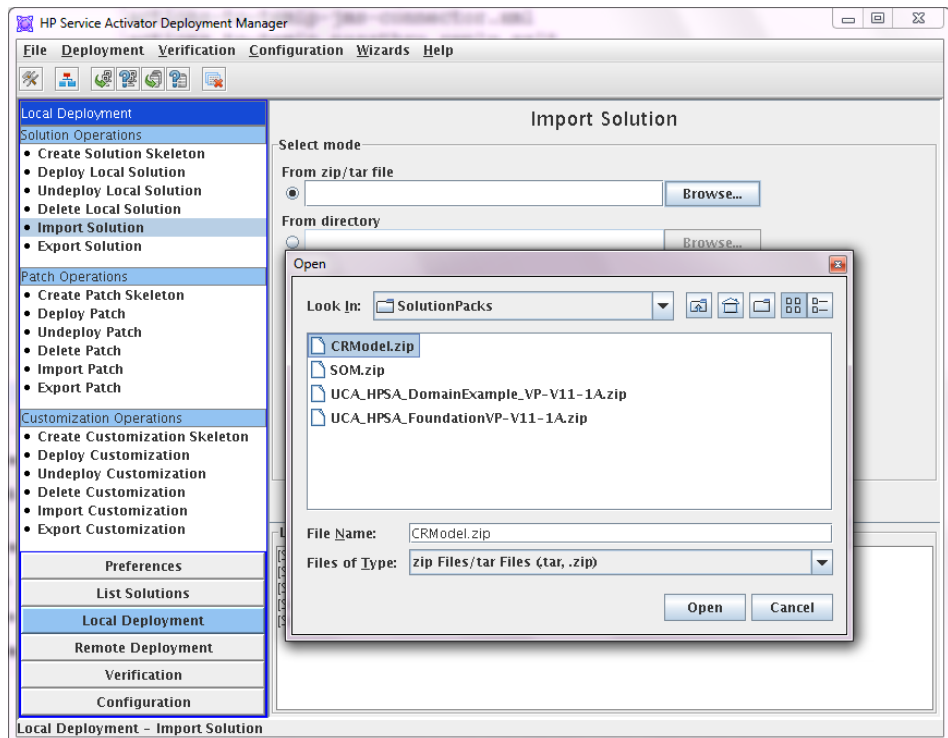
    <Decorations attributeName="status">
      <Decoration
attributeValue="Warning">images/warningLarge.png
      </Decoration>
      <Decoration
attributeValue="Failed">images/critical.png</Decoration>
    </Decorations>
  </Icon>
  <Text>
    <Color>0 0 0</Color>
    <Font>SansSerif</Font>
    <Size>10</Size>
    <Emphasis>plain</Emphasis>
    <DisplayedName>[${name}]</DisplayedName>
  </Text>
  <GetNeighbors automatic="true" level="20">
    <Queries>
      <Query><![CDATA[START startNode = node({nodeID})
MATCH (startNode)-[relationship]->(endNode) RETURN
startNode, relationship, endNode;]]></Query>
    </Queries>
  </GetNeighbors>
</DefaultNode>
<DefaultRelationship>
  <LineType>line</LineType>
  <SourceHead>none</SourceHead>
  <TargetHead>halffilledarrow</TargetHead>
  <DisplayedName>${Type}</DisplayedName>
</DefaultRelationship>
</Profile>

```

3. Restart UCA-EBC server.

2.3.2 HP Service Activator

- HP Service Activator version 6.2 - V62-1A and the latest patches
- Oracle 11g or Postgres Plus Advanced Server 9.2. The database can be installed on the same server or can be accessed remotely (but it must be located in the same sub network). You can also use an existing database that is used by another application. In that case, you need to create a new database user (if Oracle is used) or a new database instance (if Postgres Plus Advanced Server is used) for the exclusive use by Service Activator and UCA Automation.
- To validate the Domain Example of UCA Automation 1.1, deploy CRModel Solution of HPSA.



2.3.3 NOM

- OSS Open Mediation V7.0 and latest patches

```

# rpm -qa | grep ngossopenmediation
ngossopenmediation-7.0.0-RHEL5.noarch
# nom admin --list-ip-in-container
DEPLOYED          nom-basic-smx-components
  
```

```
DEPLOYED          smx-basic-components
```

- **UCA for Event Based Correlation Channel Adapter V3.1**

```
# rpm -qa | grep UCA-EBCCA
UCA-EBCCA-V3.1-00A.noarch
# nom admin --list-ip-in-container | grep uca-ebc-ca
DEPLOYED          uca-ebc-ca-3.1
```

- **The TeMIP Channel Adapter: if your solution involves TeMIP**

```
# rpm -qa | grep -i ngosstemip-ca
ngosstemip-ca-2.0.0-RHEL5.noarch
# nom admin --list-ip-in-container | grep temip-ca-20
DEPLOYED          temip-ca-20
```

2.3.4 Configure TeMIP Channel Adaptor for UCA Automation

Note

This configuration is optional.

1. **Validate the TeMIP configuration in the file.**
2. **Edit the `${NOM_INSTANCE}/ips/temip-ca-20/etc/conf/TeMIP_configuration.dynamic.xml` file.**

For example, if the TeMIP Director is configured on the `ossdemol.ind.hp.com` host, the file should contain the following configuration:

```
<TeMIPDirectorEntity>.temip.ossdemol temip</TeMIPDirectorEntity>
```

3. **Edit the `${NOM_INSTANCE}/ips/temip-ca-20/etc/conf/TeMIP_configuration.dynamic.xml` to add the custom attributes required for UCA Automation 1.1 within the tags.**

```
<CustomAttributes>
.....
</CustomAttributes>
<CustomAttribute>
  <Attribute>Action</Attribute>
  <Datatype>XmlString</Datatype>
</CustomAttribute>

<CustomAttribute>
  <Attribute>Actionidlist</Attribute>
  <Datatype>XmlString</Datatype>
</CustomAttribute>

<CustomAttribute>
  <Attribute>Actionstatus</Attribute>
  <Datatype>XmlString</Datatype>
</CustomAttribute>

<CustomAttribute>
  <Attribute>Evp</Attribute>
  <Datatype>XmlString</Datatype>
</CustomAttribute>

<CustomAttribute>
  <Attribute>Evpscenario</Attribute>
  <Datatype>XmlString</Datatype>
</CustomAttribute>

<CustomAttribute>
```

```

    <Attribute>Outputparameters</Attribute>
    <Datatype>XmlString</Datatype>
</CustomAttribute>

<CustomAttribute>
    <Attribute>Problem</Attribute>
    <Datatype>XmlString</Datatype>
</CustomAttribute>

<CustomAttribute>
    <Attribute>Rawresult</Attribute>
    <Datatype>XmlString</Datatype>
</CustomAttribute>

<CustomAttribute>
    <Attribute>Resourceinstance</Attribute>
    <Datatype>XmlString</Datatype>
</CustomAttribute>

<CustomAttribute>
    <Attribute>Serviceinstance</Attribute>
    <Datatype>XmlString</Datatype>
</CustomAttribute>

<CustomAttribute>
    <Attribute>Taskid</Attribute>
    <Datatype>XmlString</Datatype>
</CustomAttribute>

<CustomAttribute>
    <Attribute>Initiator</Attribute>
    <Datatype>XmlString</Datatype>
</CustomAttribute>

<CustomAttribute>
    <Attribute>Originatingfms</Attribute>
    <Datatype>XmlString</Datatype>
</CustomAttribute>

```

4. Edit the `${NOM_INSTANCE}/ips/temp-ca-20/etc/actions.to-temp.ao.request.uca-to-tws.xslt` file and add the customAttributes required for UCA Automation 1.1 in the `<!--Specific output nodes > tag.`

```

<!-- Specific output nodes -->
<xsl:template name="Action">
    <xsl:call-template name="simple_node">
        <xsl:with-param name="input_node"
select="command/entry[key='Action']" />
        <xsl:with-param name="output_node_name"
select="'Action'" />
    </xsl:call-template>
</xsl:template>
<xsl:template name="Actionidlist">
    <xsl:call-template name="simple_node">
        <xsl:with-param name="input_node"
select="command/entry[key='Actionidlist']" />
        <xsl:with-param name="output_node_name"
select="'Actionidlist'" />
    </xsl:call-template>
</xsl:template>
<xsl:template name="Actionstatus">
    <xsl:call-template name="simple_node">
        <xsl:with-param name="input_node"
select="command/entry[key='Actionstatus']" />

```

```

        <xsl:with-param name="output node name"
select="'Actionstatus'" />
    </xsl:call-template>
</xsl:template>
<xsl:template name="Evp">
    <xsl:call-template name="simple_node">
        <xsl:with-param name="input node"
select="command/entry[key='Evp']" />
        <xsl:with-param name="output_node_name"
select="'Evp'" />
    </xsl:call-template>
</xsl:template>
<xsl:template name="Evpscenario">
    <xsl:call-template name="simple_node">
        <xsl:with-param name="input_node"
select="command/entry[key='Evpscenario']" />
        <xsl:with-param name="output_node_name"
select="'Evpscenario'" />
    </xsl:call-template>
</xsl:template>
<xsl:template name="Outputparameters">
    <xsl:call-template name="simple node">
        <xsl:with-param name="input_node"
select="command/entry[key='Outputparameters']" />
        <xsl:with-param name="output node name"
select="'Outputparameters'" />
    </xsl:call-template>
</xsl:template>
<xsl:template name="Problem">
    <xsl:call-template name="simple node">
        <xsl:with-param name="input node"
select="command/entry[key='Problem']" />
        <xsl:with-param name="output node name"
select="'Problem'" />
    </xsl:call-template>
</xsl:template>
<xsl:template name="Rawresult">
    <xsl:call-template name="simple_node">
        <xsl:with-param name="input node"
select="command/entry[key='Rawresult']" />
        <xsl:with-param name="output_node_name"
select="'Rawresult'" />
    </xsl:call-template>
</xsl:template>
<xsl:template name="Resourceinstance">
    <xsl:call-template name="simple_node">
        <xsl:with-param name="input_node"
select="command/entry[key='Resourceinstance']" />
        <xsl:with-param name="output_node_name"
select="'Resourceinstance'" />
    </xsl:call-template>
</xsl:template>
<xsl:template name="Serviceinstance">
    <xsl:call-template name="simple node">
        <xsl:with-param name="input_node"
select="command/entry[key='Serviceinstance']" />
        <xsl:with-param name="output_node_name"
select="'Serviceinstance'" />
    </xsl:call-template>
</xsl:template>
<xsl:template name="Taskid">
    <xsl:call-template name="simple_node">

```

```

        <xsl:with-param name="input node"
select="command/entry[key='Taskid']" />
        <xsl:with-param name="output_node_name"
select="'Taskid'" />
        </xsl:call-template>
    </xsl:template>
    <xsl:template name="Initiator">
        <xsl:call-template name="simple_node">
            <xsl:with-param name="input_node"
select="command/entry[key='Initiator']" />
            <xsl:with-param name="output_node_name"
select="'Initiator'" />
        </xsl:call-template>
    </xsl:template>
    <xsl:template name="Originatingfms">
        <xsl:call-template name="simple node">
            <xsl:with-param name="input_node"
select="command/entry[key='Originatingfms']" />
            <xsl:with-param name="output node name"
select="'Originatingfms'" />
        </xsl:call-template>
    </xsl:template>

```

5. Add the following entry under the <!-- CREATE: 27 items --> tag.

```

<!-- ***** -->
<xsl:if test="($request type='Create')">

<xsl:call-template name="Action" />
    <xsl:call-template name="Actionidlist" />
    <xsl:call-template name="Actionstatus" />
    <xsl:call-template name="Evp" />
    <xsl:call-template name="Evpscenario" />
    <xsl:call-template name="Outputparameters" />
    <xsl:call-template name="Problem" />
    <xsl:call-template name="Rawresult" />
    <xsl:call-template name="Resourceinstance" />
    <xsl:call-template name="Serviceinstance" />
    <xsl:call-template name="Taskid" />
    <xsl:call-template name="Originatingfms" />
    <xsl:call-template name="Initiator" />

```

6. Add the following entry under <!-- SET: 82 items of Attribute List -->

```

<xsl:variable name="of_type_timestamp"
select="('Acknowledgement_Time_Stamp',...,
        'Pb',
        'Parents',
        'Children',
        'Action',
        'Actionidlist',
        'Actionstatus',
        'Evp',
        'Evpscenario',
        'Outputparameters',
        'Problem',
        'Rawresult',
        'Resourceinstance',
        'Serviceinstance',
        'Taskid',
        'Initiator',
        'Originatingfms')"/>

```

7. Redeploy the TeMIP channel adaptor.

```
#nom admin --undeploy-ip-in-container temip-ca-20
#nom_admin --deploy-ip-in-container temip-ca-20
```

2.3.5 JAVA

If UCA-EBC and HPSA reside on the same server then use Java SE 6 update 37 JDK or a later version(as HPSA does not support JAVA 1.7).

2.3.6 Configure TeMIP 6.2L

Note

This configuration is optional.

When TeMIP 6.2L is used as NMS, perform the following:

1. Install and configure TeMIPV62L with the latest available patches.
For more details, refer to the *TeMIP Installation Guide for Linux*.
2. Install and configure TeMIP Web Services with the latest patches.
For more details, refer to the *TeMIP WebServices Installation and Administration Guide*.
3. Set **TeMIP Web Services Security**' level to none.
4. Run the following command to update the TeMIP dictionary with custom attributes required for UCA EBC 3.1.

```
/usr/opt/temip/bin/temip_ah_user_defined_attr -project TPD
```

5. Run the following command to update the TeMIP dictionary with custom attributes required for UCA Automation 1.1.

```
/usr/opt/temip/bin/temip_ah_user_defined_attr -project TND
```

2.3.7 Configure TeMIP when used as NMS

Note

This configuration is optional.

When TeMIP is used as an NMS, configure TeMIP using the following procedure.

1. Enter the TeMIP management and run the following commands.

```
$ manage
TeMIP Framework (V6.2.0)
```

```
TeMIP> create domain uca_dom
Domain tfrsoll_ns:.uca_dom
On director: tfrsoll ns:.temip.tfrsoll director
AT Mon 11 Mar 2013 06:54:49

Entity successfully created.
```

```
TeMIP> create oper uca_network assoc domain uca_dom
OPERATION CONTEXT tfrsoll_ns:.uca network
On director: tfrsoll ns:.temip.tfrsoll director
AT Mon 11 Mar 2013 06:55:29

Operation Context successfully created
```



```
TeMIP> create oper uca pbalarm assoc domain uca dom
OPERATION_CONTEXT tfrsoll_ns:.uca_pbalarm
On director: tfrsoll_ns:.temip.tfrsoll_director
AT Mon 11 Mar 2013 06:56:07

Operation Context successfully created
```

```
TeMIP> register oper uca network
OPERATION_CONTEXT tfrsoll_ns:.uca_network
On director: tfrsoll_ns:.temip.tfrsoll_director
AT Mon 11 Mar 2013 06:56:22

Registration successful.
```

```
TeMIP> register oper uca pbalarm
OPERATION_CONTEXT tfrsoll_ns:.uca_pbalarm
On director: tfrsoll_ns:.temip.tfrsoll_director
AT Mon 11 Mar 2013 06:56:36

Registration successful.
```

2.4 Web client

- Mozilla Firefox 32
- Google Chrome 37

UCA Automation solution pack

This chapter includes the procedures to install the following components:

- UCA Automation Solution Pack
- HPSA Foundation Value Pack
- UCA EBC Foundation Value Pack
- UCA Automation UI
- NOM Channel Adapters

3.1 Installing UCA Automation solution

The UCA Automation solution is delivered as an RPM file named:

```
UCA_Automation-V1.1-REV_A.noarch.rpm
```

To install the package, perform the following operations as a **root** user:

1. Copy the file `UCA_Automation-V1.1-REV_A.noarch.rpm` to a RHEL system, and store it in a directory, for example, directory `/tmp`.
2. Run the following command to install the package.

```
# rpm -ivh UCA_Automation-V1.1-REV_A.noarch.rpm
```

It installs the package under `/opt/UCA_Automation` directory. The following directories are created.

- TeMIP_Integration
 - TEMIPTFRLIN_00172.tar
 - TEMIPTFRLIN_00172.text
- UCA_Automation_ChannelAdapters
 - uca-autoconsole-ca-2.0.0-L.tar
 - uca-hpsa-ca-2.0.0-L.tar
- UCA_Automation_HPSA_VPs
 - UCA_HPSA_DomainExample_VP-V11-1A.zip
 - UCA_HPSA_FoundationVP-V11-1A.zip
- UCA_Automation_UCA_VPs
 - UCA_Automation_DomainExample_UCA_EV-vp-V1.1-1A.zip
 - UCA_Automation_Foundation_UCA-vp-V1.1-1A.zip
 - UCA_Automation_DomainExample_UCA_PD-vp-V1.1-1A.zip
- UCA_Automation_Documents
 - UCA-Autoconsole-CA.pdf

- UCA-HPSA-CA.pdf
- HP_UCA_Automation_1.1_Install_Guide_1.0.pdf
- HP_UCA_Automation_1.1_Admin_User_Interface_Guide_1.0.pdf
- HP_UCA_Automation_1.1_Integration_Guide_1.0.pdf

Artifact	Description
TEMIPTFRLIN_00172.tar	TeMIP Server Patch – User Defined Attributes
uca-autoconsole-ca-2.0.0-L.tar	UCA-Automation Console Channel Adapter
uca-hpsa-ca-2.0.0-L.tar	UCA-HPSA Channel Adapter
UCA_HPSA_DomainExample_VP-V11-1A.zip	HPSA example VP
UCA_HPSA_FoundationVP-V11-1A.zip	HPSA Foundation VP
UCA_Automation_DomainExample_UCA_EV-vp-V1.1-1A.zip	UCA EBC example evaluate value pack
UCA_Automation_Foundation_UCA-vp-V1.1-1A.zip	UCA EBC Foundation VP
UCA_Automation_DomainExample_UCA_PD-vp-V1.1-1A.zip	UCA EBC example PD value pack

Table 3 RPM artifacts

3. Verify if the package is installed successfully.

- a. Run the following command:

```
# rpm -qa | grep -i Automation
UCA_Automation-V1.1-REV_A.noarch
```

- b. Uninstall the packages by running the following commands:

```
rpm -ev UCA_Automation-V1.1-REV_A.noarch
```

3.2 Installing HPSA Foundation value pack

3.2.1 Deploy HPSA Foundation value pack

The HPSA foundation value pack is delivered as a ZIP file named `UCA_HPSA_FoundationVP-V11-1A.zip`.

1. As a root user, copy the ZIP file of the foundation value pack to the `${ACTIVATOR_OPT}/SolutionPacks` directory.
2. Import and deploy the Foundation Value Pack solution.
For details, refer the *Deployment Manager Guide*.
3. Make sure that the **Create Inventory Table** checkbox is selected.

For information on undeploying and deleting the HPSA solution pack, refer to the *Deployment Manager Guide*.

3.2.2 Configure HPSA Foundation value pack

1. As a **root** user, run the **config.sh** script in the `${SOLUTION_ETC}/config` directory.

The script enables the `httpsender` module in the `MWFM.xml` file of the HPSA with the web service URL hosted in the HPSA Channel Adaptor. When the HTTPsender module is enabled, it sends the responses to the Automation Console.

```
# cd /opt/OV/ServiceActivator/solutions/UCA/etc/config
# chmod +x ./config.sh
# ./config.sh
Setting up the Service Activator UCA Foundation Value
Pack...

Configuring MicroWorkFlow Manager
(/etc/opt/OV/ServiceActivator/config/mwfm.xml)...
=====
====

UCA HTTP Sender module...
Enter Host name/IP address of the web service hosted in
HPSA Channel Adaptor
[localhost] :
Enter port for web service hosted in HPSA Channel Adapter
[ 8191 ] :
8191
(Saving mwfm.xml for future reconfiguration)

/etc/opt/OV/ServiceActivator/config/mwfm.xml configured

Done setting up Service Activator Foundation Value Pack

Log file:
/var/opt/OV/ServiceActivator/log/tfrsol1/ucasp.install.0318
13 013907.log

Changes in Service Activator configuration files
may be inspected in files:
/var/opt/OV/ServiceActivator/log/tfrsol1/uca.mwfm.xml.diff

Press enter to continue...
```

The following is a snippet of the `mwfm.xml` file.

```
<Module>
  <Name>uca_http_sender</Name>
  <Class-Name>com.hp.ov.activator.mwfm.engine.module.HTTPSenderModule</Class-Name>
  <Param name="url" value="http://0.0.0.0:8191/UCAAutomation/UCAService"/>
  <Param name="connect_timeout" value="10000"/>
  <Param name="read_timeout" value="10000"/>
  <Param name="min_threads" value="1"/>
  <Param name="max_threads" value="3"/>
  <Param name="queue_name" value="httprequest"/>
  <Param name="retry_count" value="3"/>
  <Param name="retry_interval" value="40000"/>
  <Param name="queue_class" value="com.hp.ov.activator.mwfm.engine.module.Weight
edEngineQueue"/>
</Module>
```

2. Reload the configuration from the HPSA UI or restart HP Service Activator.

3.3 Installing UCA EBC Foundation value pack

The UCA Automation foundation value pack is delivered as a ZIP file named UCA_Automation_Foundation_UCA-vp-V1.1-1A.zip.

1. Copy the ZIP file of the foundation value pack to the `${UCA_EBC_INSTANCES}/valuepacks` directory.
2. Deploy the Foundation value pack.
For details, refer the *UCA for Event Based Correlation Value Pack Development Guide*.
3. Edit the `${UCA_EBC_DATA}/instances/default/conf/uca-ebc-log4j.xml` file.
4. Add the following section in the file under the root tag `<log4j:configuration>`, specifically below the comment line Detailed Traces for Value Pack Scenarios:

```
<logger
name="UCA_Automation_Foundation_UCA.requestresponse"
additivity="false">
    <level value="TRACE" />
    <appender-ref ref="CONSOLE" />
    <appender-ref ref="FILE" />
</logger>

<logger name="UCA_Automation_DomainExample_UCA_EV.evaluate"
additivity="false">
    <level value="TRACE" />
    <appender-ref ref="CONSOLE" />
    <appender-ref ref="FILE" />
</logger>
```

3.3.1 Configure UCA Automation UI

1. Edit the `UCAAutomation.properties` file in the `${UCA_EBC_INSTANCES}/deploy/UCA_Automation_Foundation_UCA-V1.1-1A/conf` directory.
2. Update the localhost with UCA EBC Server hostname.

```
ucaebc tomsawyer port=http://localhost:8888/graphdisplay/?u
sername=root&nodeId=0&profile=ucaatm
```
3. Update the database.
 - o If you have an enterprise database Postgres, use the following configuration.

```
DB_DRIVER=org.postgresql.Driver
DB_URL=jdbc:postgresql://<db-host>:<db-port>/<db>
DB_USER=<db-user>
DB_PASSWORD=<db-user-password>
```
 - o If you have Oracle 11g database, use the following configuration.

```
DB_DRIVER=oracle.jdbc.driver.OracleDriver
DB_URL=jdbc:oracle:thin:@<db-host>:<db-port>:<db>
DB_USER=<db-user>
DB_PASSWORD=<db-user-password>
```
4. Edit the `ExternalActionConfig.xml` file available in the `${UCA_EBC_INSTANCES}/deploy/UCA_Automation_Foundation_UCA-V1.1-1A/conf` directory and edit the following line with the UCA EBC server host name and port:

```
<consoleurl>  
http://localhost:8888/UCA_Automation_Foundation_UCA-V1.1-  
1A-UCAAutomation/UCAService  
</consoleurl>
```

5. If TeMIP is used as NMS, start the UCA_Automation_Foundation_UCA value pack.
6. If TeMIP is not the NMS, before starting the UCA_Automation_Foundation_UCA value pack, delete the mediation flow in UCA_Automation_Foundation_UCA value pack.

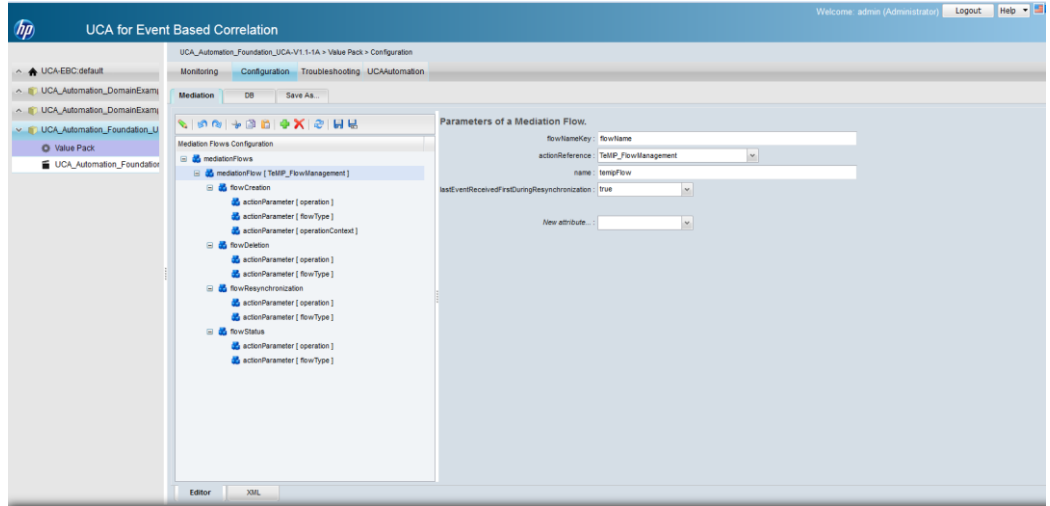


Figure 1 Deleting the Mediation Flow from UCA_Automation_Foundation_UCA value pack

3.4 Installing UCA Automation UI

3.4.1 Deploy UCA Automation UI

The UCA Automation UI is embedded in UCA Automation Foundation value pack and is delivered as a ZIP file.

Deploy and start the UCA Automation Foundation value pack to view the UCA Automation UI.

Chapter 4

UCA Automation Orchestrator

The UCA Automation 1.1 release provides an Eclipse plug-in to create and deploy UCA Automation Orchestrator.

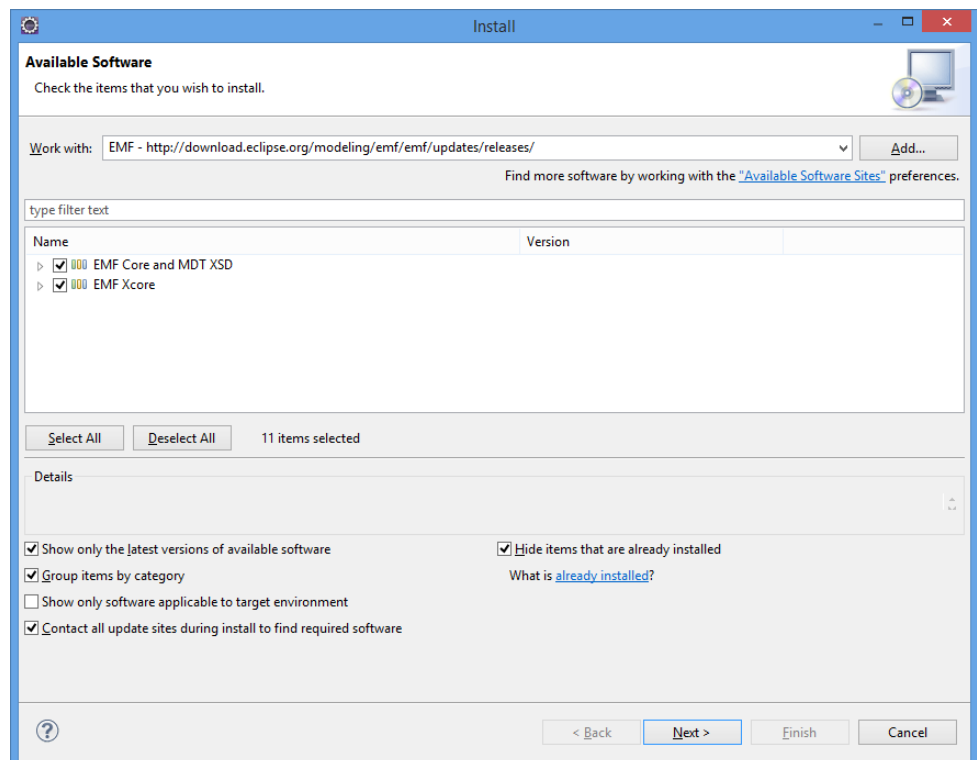
4.1 Prerequisites for installing Orchestrator

Eclipse Kepler version 4.3.2

4.2 Install UCA Automation Orchestrator

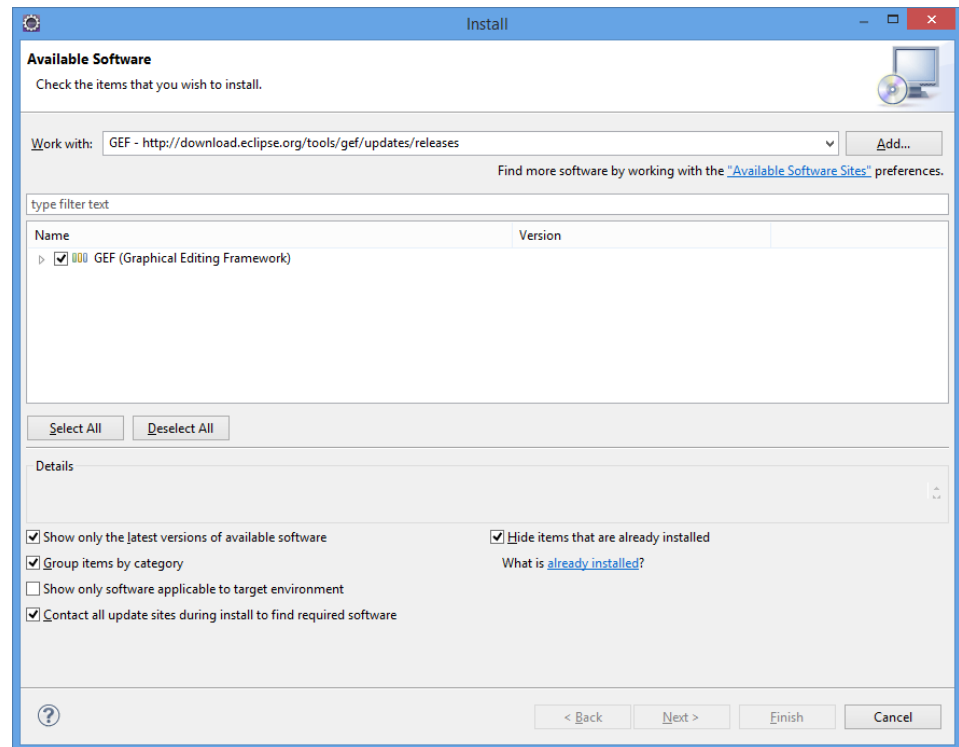
4.2.1 Install EMF plug-in

1. Open Eclipse.
2. Select **Help** -> **Install New Software**.
3. Install EMF from the following site:
`http://download.eclipse.org/modeling/emf/emf/updates/releases/`
4. Select **All** and click **NEXT** to install the software.



4.2.2 Install GEF plug-in

1. Open Eclipse.
2. Select **Help** -> **Install New Software**.
3. Install GEF from the following site:
<http://download.eclipse.org/tools/gef/updates/releases>
4. Select **All** and click **NEXT** to install the software.



4.2.3 Install Windows builder

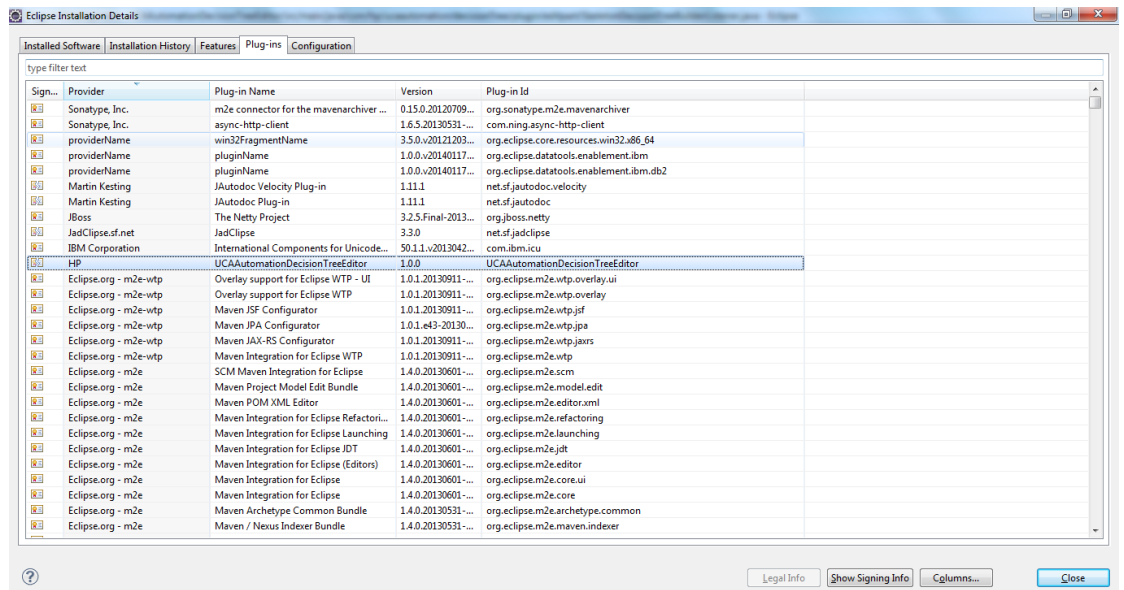
1. Install the Windows builder directly from the following location:
<http://download.eclipse.org/windowbuilder/WB/release/R201309271200/4.3/>
2. Place the `UCAAutomationOrchestrator.jar` file in the `dropins` folder of the Eclipse.
3. Restart the Eclipse.

Note

If the `UCAAutomationOrchestrator.jar` file is updated, after placing the updated JAR file in the `dropins` folder of Eclipse, restart Eclipse with a `-clean` option.

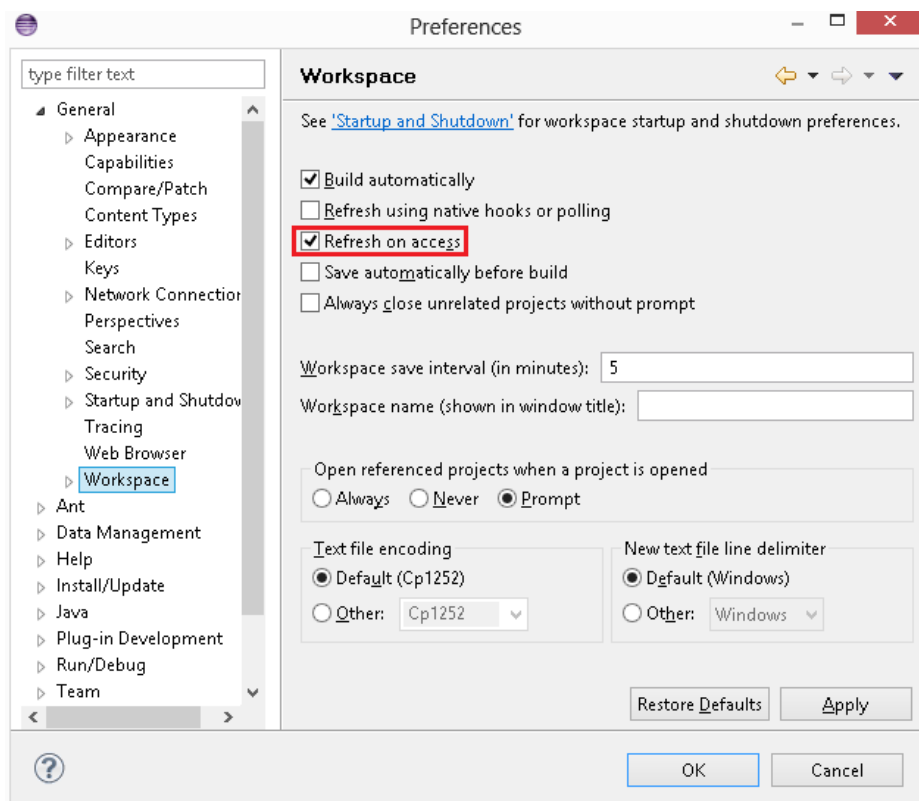
4. To verify whether the plug-in is installed, select **Help** -> **About Eclipse** -> **Installation details** -> **Plugins**.

The following **Plug-ins** tab in the **Eclipse Installation Details** window contains the plug-in name.

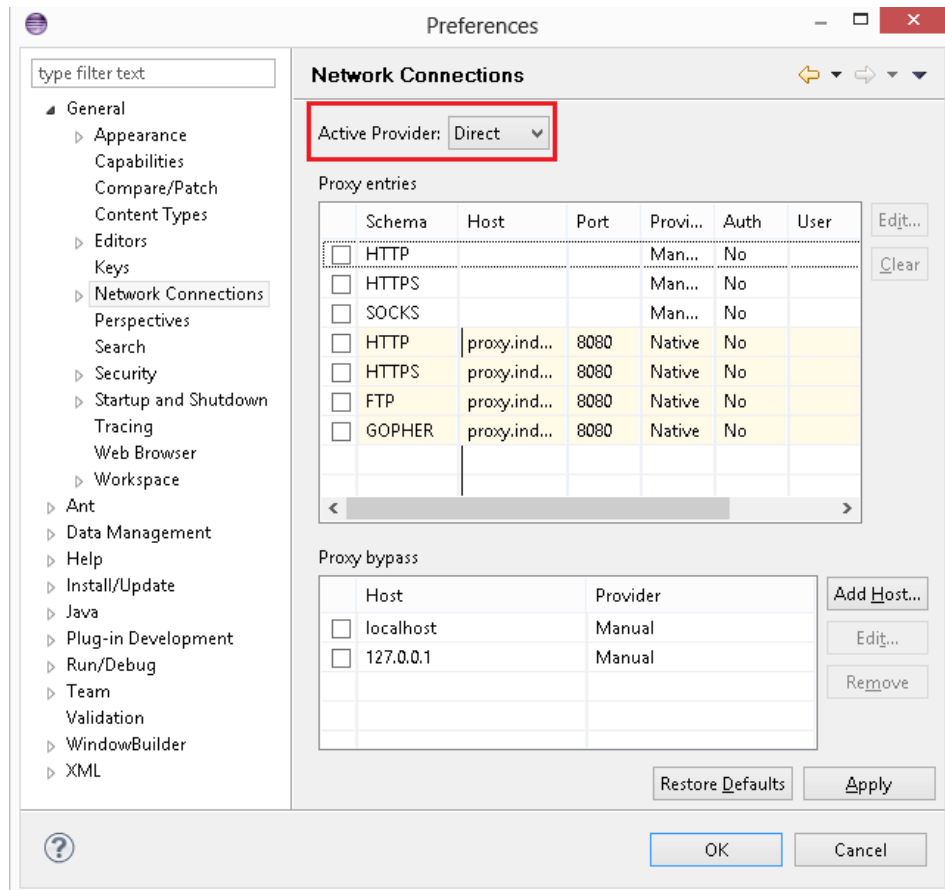


4.3 Configure Orchestrator

1. After loading Eclipse, select **Window -> Preferences -> General -> Workspace**.
2. Select the checkbox for **Refresh on access**.



3. Configure the Active Provider in Eclipse to improve the upload performance of the decision tree.
 - a. Select **Window -> Preferences -> General -> Network Connections**.
 - b. Set the **Active Provider** to **Direct**.



4.4 Decision tree command line utility

Edit the `GraphDBUtilities.properties` file available under the `${UCA_AUTOMATION_CONSOLE_HOME}/utilities/DecisionTree/conf/` directory.

```
#UCA Automation Inventory database connection details
#Oracle jdbc driver :
#oracle.jdbc.pool.OracleConnectionPoolDataSource
#Oracle url : jdbc:oracle:thin:@<hostname>:<port>:<service>
#Postgres jdbc driver : com.edb.Driver
#Postgres url : jdbc:edb:@<hostname>:<port>:<service>
inventory.db.driver=com.edb.Driver
inventory.db.url=jdbc:edb://<hostname>:<port>/<service>
inventory.db.user=<username>
inventory.db.password=<password>

#Neo4j database connection details
neo4j.db.protocol=http
neo4j.db.host=localhost
neo4j.db.port=7474
neo4j.db.db=db
neo4j.db.data=data
#enables batch transaction of inserts into neo4j db
org.neo4j.rest.batch_transaction=true
#enables http streaming
org.neo4j.rest.stream=true
```

The following is a snippet of the `GraphDBUtilities.properties` file.

```

#UCA Automation Inventory database connection details
#Oracle jdbc driver : oracle.jdbc.pool.OracleConnectionPoolDataSource
#Oracle url : jdbc:oracle:thin:@<hostname>:<port>:<service>
#Postgres jdbc driver : com.edb.Driver
#Postgres url : jdbc:edb:@<hostname>:<port>:<service>
inventory.db.driver=com.edb.Driver
inventory.db.url=jdbc:edb://localhost:5444/postgres
inventory.db.user=hpsa61
inventory.db.password=hpsa61

#Neo4j database connection details
neo4j.db.protocol=http
neo4j.db.host=localhost
neo4j.db.port=7474
neo4j.db.db=db
neo4j.db.data=data
#enables batch transaction of inserts into neo4j db
org.neo4j.rest.batch_transaction=true
#enables http streaming
org.neo4j.rest.stream=true

```

Properties	Description
inventory.db.driver	Database driver name.
inventory.db.url	The connection URL.
inventory.db.user	The login user name of database.
inventory.db.password	The login password associated with the user name
neo4j.db.protocol	By default it is http.
neo4j.db.host	By default it is localhost. This is the hostname/IP address of the server where neo4j is hosted.
neo4j.db.port	By default it is 7474. The port number where neo4j is hosted.
neo4j.db.db	By default it is db
neo4j.db.data	By default it is data
org.neo4j.rest.batch_transaction	Enables batch transaction of inserts into neo4j
org.neo4j.rest.stream	Enables http streaming

Table 4 Neo4j configuration descriptors

Install NOM channel adaptors

5.1 HPSA channel adaptor

To install and deploy the HPSA Channel Adaptor, follow the instructions in the UCA HPSA CA Main Release Guide.

Edit the `config.properties` file in the `${NOM_INSTANCE}/ips/uca-hpsa-ca-20/etc`.

```
# HPSA connectivity settings
hpsa.host=0.0.0.0
hpsa.port=<HPSA port>
hpsa.userid=<HPSA user with StartJob privileges>
hpsa.password=<password for the above HPSA user>

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

# UCA-Automation response handler connectivity settings
hpsa.uca-automation.sync-service.host=0.0.0.0
hpsa.uca-automation.sync-service.port=8191
```

The following is a snippet of the `config.properties`:

```
# HPSA connectivity settings
hpsa.host=0.0.0.0
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=0.0.0.0
hpsa.uca-automation.sync-service.port=8191
```

Descriptor	Description
<code>hpsa.host</code>	Hostname /IP address of the server where HPSA is hosted.
<code>hpsa.port</code>	The listening port of HPSA
<code>hpsa.userid</code>	The login user name of HPSA. The user has the permission to start and stop a job, as well as check the status.
<code>hpsa.password</code>	The login password associated with the user name
<code>hpsa.uca-automation.sync-service.host</code>	The default value is 0.0.0.0. This is the hostname/IP address of the internal web service being hosted by HPSA CA
<code>hpsa.uca-automation.sync-service.port</code>	The default value is 8191. The listening port number of the internal web service hosted by CA.
<code>hpsa.controller.workflow.name</code>	The name of the HPSA foundation workflow to be invoked in HPSA. You cannot change the value.

Table 5 HPSA CA config descriptors

5.2 UCA Automation console channel adaptor

To install and deploy the Automation Console Channel Adaptor, follow the instructions in the *UCA Autoconsole CA Main Release Guide*.

Edit the `config.properties` file in the `${NOM_INSTANCE}/ips/uca-autoconsole-ca-20/etc` directory.

```
uca.uca-automation.host=0.0.0.0
uca.uca-automation.port=12500
uca.console.service=UCA Automation Foundation UCA-V1.1-1A-
UCAAutomation/UCAService
uca.console.host=localhost
uca.console.port=<uca-ui.properties - ucaui.gui.port>
```

The following snippet from the `config.properties` file contains sample values:

```
uca.uca-automation.host=0.0.0.0
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
```

Descriptor	Description
<code>uca.uca-automation.host</code>	Host name or the IP address of Automation console CA where the internal webservice is hosted. The default value is 0.0.0.0.
<code>uca.uca-automation.port</code>	Listening port of the internal Automation console CA web service. The default value is 12500.
<code>uca.console.service</code>	Name of the automation console web service.
<code>uca.console.host</code>	The Hostname /IP address of the server where the UCA Automation console is hosted.
<code>uca.console.port</code>	The listening port of the UCA Automation console. See section 6.2, <code>uca-ui.properties</code> - <code>ucaui.gui.port</code>

Table 6 UCA Console CA config descriptors

UCA Automation licensing

A 60-day Instant-On license is installed by default when UCA-EBC Automation Foundation value pack is started. This license activates all features of the product for a trial period. After the expiration of the trial period, an extended evaluation or a commercial license is needed to continue using the product.

For any questions related with licensing, please contact the UCA Automation product management team.

6.1 Get a UCA Automation license

You need a license key to use the UCA Automation software. Licensing is managed by AutoPassJ, which is automatically installed with UCA for EBC. You must obtain a license key to continue using the product after the 60-day trial period.

The following is the process for getting a license key:

1. Log in as a system administrator of the product and access the Webware web site (www.webware.hp.com).
2. Download the perpetual license to use the product.

To request perpetual license keys, you should have the following information:

- Entitlement Certificate—Contains the HP product number and order number (Entitlement order number).
 - Contact information of the license owner—Details of your company or organization.
3. (Optional) Contact the HP Password Center via fax, email, or phone.

For more information, refer to the Password Request Form and the License Entitlement Certificate. To get product licenses, you should have the License Entitlement Certificate.

6.2 License policy

When the UCA-EBC Automation Foundation Value Pack for EBC Server starts, the system runs a license check for UCA Automation feature.

To activate this feature, you should have a valid license key. The UCA-EBC Automation Foundation value pack does not start if you do not have a valid license key for the UCA Automation feature.

The following table shows the link between UCA Automation product part numbers and the features:

Product part number	Description	Enabled UCA Automation features
JK461AAE	HP UCA Automation Foundation Prod E-LTU	HP UCA Automation Foundation Prod E-LTU HP OSS UCA Expert Production HP OSS UCA Expert DB HP Service Activator Tier X ELTU
JK462AAE	HP UCA Automation Foundation Non-Prod E-LTU	HP UCA Automation Foundation Non-Prod E-LTU HP OSS UCA Expert Production HP OSS UCA Expert DB HP Service Activator Tier X ELTU

When you activate the UCA Automation Instant-On feature, UCA Automation automatically generates a corresponding license key which is added to the `license.txt` file available under the `${UCA_EBC_DATA}/instances/<instance name>/licenses` folder. By default the directory path is `/var/opt/UCA-EBC/instances/default/licenses`.

This `license.txt` file contains all license keys. For information on installing the licenses, refer to the UCA-EBC and HP Service Activator *Installation Guides*.

6.3 Use Webware website for product licenses

The following procedure discusses how to obtain your product licenses.

- Go to www.webware.hp.com.
The HP Licensing for Software website opens.
- Click the **Sign In** button.
The **HP Passport Sign-in** page appears.
- Sign in using the user ID and password associated with your HP Passport.
If you do not have an HP Passport account, request for one by clicking the **New users - Please register** link.
- Click **Sign In**.
The home page of the HP Licensing for Software website opens.
- Enter your entitlement order number in the **Entitlement order number (EON)** text box.
Your order number is displayed on the License Entitlement Certificate (HP Order Number).
- Click **Go**.
- In the Entitlement column, select the checkboxes for the products for which you want license keys.
- Click **Activate**.
- For each product you selected, enter the number of Licenses to Use (LTUs) in the **Qty** field.

The number of licenses is limited by the total number of LTUs available for the order.

10. For each product you selected, enter the required details in the **Target** field.

It includes the following: Server host name, IP address for the system where the software is installed, and so on.

11. Click **Finish**.

A confirmation message appears stating that an email containing the license keys has been sent to you. The page also displays the license keys and provides links for emailing and displaying them along with the information regarding the product activation.

6.4 Install license keys for UCA Automation

The following procedure discusses how to install license keys for UCA Automation.

1. After you get the UCA Automation license key, copy the license key to the `license.txt` file available under the `${UCA_EBC_DATA}/instances/<instance name>/licenses` folder.

The default location is `/var/opt/UCA-EBC/instances/default/licenses`.

If you want to copy more than one license key to the `license.txt` file, append them to the `license.txt` file one after the other until all license keys are copied to the `license.txt` file.

2. Restart the UCA for EBC Server to apply the changes made to the `license.txt` file.

6.5 Remove license keys for UCA Automation

The following procedure discusses how to remove license keys for UCA Automation.

1. Remove the license keys from the `license.txt` file located in the `${UCA_EBC_DATA}/instances/<instance name>/licenses` folder.

The default location is `/var/opt/UCA-EBC/instances/default/licenses`.

Code signing

This software product from HP is digitally signed and accompanied by Gnu Privacy Guard (GnuPG) signatures. HP strongly recommends using signature verification on its products, but there is no obligation. Customers have the choice of running this verification or not as per their IT Policies.

7.1 Install and configure Gnu Privacy Guard (GnuGP)

If you do not have GnuGP installed, you should download and install GnuGP. For information about obtaining and installing GnuGP, see <http://www.gnupg.org>

Before verifying the signatures delivered on the HP Service Activator DVD, you should configure GnuGP for accepting the HP signature. Use the following procedure to configure GnuGP:

1. Log into your system.
2. Get the HP public key from following location:

```
https://h20392.www2.hp.com/portal/swdepot/displayProductInfo.do?productNumber=HPLinuxCodeSigning
```

3. Save the key as `hpPublicKey.pub`.
4. Import the key into GnuPG by running the following command

```
gpg --import hpPublicKey.pub
```

7.2 Verify authenticity and integrity in RHEL 6.4

This section explains the procedure to verify the signatures of the software packages to assess the integrity of the software before installation.

In the command prompt, go to the home directory on the DVD and run the following command:

```
gpg --verify UCA_Automation-V1.0-REV_A.noarch.rpm.sig  
UCA_Automation-V1.0-REV_A.noarch.rpm
```

You should get the following output from the **gpg** command:

```
gpg: Good signature from "Hewlett-Packard Company (HP Code  
signing Service) "
```

Glossary

Term	Description
UCA	Unified Correlation Analyzer
EBC	Event Based Correlation
IP	Installation Package for OSS Open Mediation V6.2
JDK	Java Development Kit
JMS	Java Messaging Service
JNDI	Java Naming and Directory Interface
JRE	Java Runtime Environment
Inference Engine	Process that uses a Rete algorithm
DRL	Drools Rule file
XML	Extensible Markup Language
XSD	Schema of an XML file, describing its structure
NOM	NextGen OSS Open Mediation
CA	NOM Channel Adapter
PPAS	Postgres Plus Advanced Server