hp Unified Correlation Analyzer

Unified Correlation Analyzer
for
Event Based Correlation

Version 3.1

Release Notes

Edition: 1.0

For the Operating Systems:
Linux (RHEL 5.8 & 6.3)
HP-UX (11.31)
Windows© (for development toolkit)

April 2014

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

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.

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

Preface 6

Chapter 1 8

New features provided by this release (V3.1) 8

1.1 Orchestration feature 8
1.2 UCA for EBC Channel Adapter enhancements 8
1.3 The UCA for EBC GUI URL has changed 9
1.4 UCA for EBC GUI can be made optional 10
1.5 Value pack specific GUI extension 10
1.6 UCA for EBC GUI extras 10
1.7 UCA for EBC GUI filter builder utility 10
1.8 UCA for EBC GUI upload/remove value pack features 11
1.9 Tuning a UCA for EBC Server instance JVM boot options 11
1.10 Graphical Display of the UCA for EBC Topology 11
1.11 GUI configuration of Orchestration feature 12
1.12 Performance enhancements 12
1.13 Specific documentation of the TeMIP Integration 12
1.14 Data Base Alarm Forwarding 12
1.15 Data Base Alarm Flow 12
1.16 REST API to access Alarm Data Base 12
1.17 Generic Event interface 13
1.18 Data Base Generic Event Forwarding 13
1.19 uca-ebc-admin command-line tool enhancements 13
1.20 Support for multiple filter files 13
1.21 Value pack specific GUI extension (URL service path) 14
1.22 Value pack specific GUI extension (URL parameters) 14
1.23 Value pack specific GUI extension (single sign-on) 14
1.24 “Impermanent” topology data base for Topology based value pack JUnit tests 14
1.25 Bug fixing 15

Chapter 2 16

Noticeable changes compared to previous released version (V3.0) 16

2.1 Changes to the AbstractJunitIntegrationTest class 16
2.2 Changes to the value pack build file: build.xml 16
2.3 Changes to the value pack configuration file: ValuePackConfiguration.xml 17

Chapter 3 19

Fixed Problems 19

Chapter 4 22

Known Problems 22

Figures

Figure 1 - Setting the CXF component thread pool size in the actions-to-nom.xml file 9
Figure 2 - Setting the Camel Context thread pool size in the actions-to-nom.xml file 9
Figure 3 - Example of a topology view around a Point Of Interest 11
Figure 4 - Referencing multiple filter files inside a ValuePackConfiguration.xml file 14
Figure 5 - Setting the lastEventReceivedFirstDuringResynchronization attribute in the ValuePackConfiguration.xml file 17
Figure 6 - Workaround for defect CR#11451 23

Tables

Table 1 - Software versions 6
Table 2 - Fixed Problems in UCA for EBC V3.1 21
Table 3 - Known Problems 23

Preface

These Release Notes describe critical information related to the HP UCA for Event Based Correlation product.

Product Name: Unified Correlation Analyzer for Event Based Correlation

Product Version: 3.1

Kit Version: V3.1

Please read this document before installing or using this Software.

Intended Audience

Here are some recommendations based on possible reader profiles:

- Solution Developers
- 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 for Event Based Correlation Server Version 3.1	• HP-UX 11.31 for Itanium • Red Hat Enterprise Linux Server release RHEL 5.8 & 6.3
UCA for Event Based Correlation Channel Adapter Version 3.1	• HP-UX 11.31 for Itanium • Red Hat Enterprise Linux Server release RHEL 5.8 & 6.3
UCA for Event Based Correlation Topology Extension Version 3.1	• HP-UX 11.31 for Itanium • Red Hat Enterprise Linux Server release RHEL 5.8 & 6.3
UCA for Event Based Correlation Software Development Kit Version 3.1	• Windows XP / Vista • Windows Server 2007 • Windows 7 • Red Hat Enterprise Linux Server release RHEL 5.8 & 6.3

Table 1 - Software versions

Typographical Conventions

Courier Font:

- Source code and examples of file contents.
- Commands that you enter on the screen.
- Pathnames
- Keyboard key names

Italic Text:

- Filenames, programs and parameters.
- The names of other documents referenced in this manual.

Bold Text:

- To introduce new terms and to emphasize important words.

Associated Documents

HP UCA for Event Based Correlation - Installation Guide

HP UCA for Event Based Correlation - Administration, Configuration, and Troubleshooting Guide

HP UCA for Event Based Correlation - Reference Guide

HP UCA for Event Based Correlation - Topology Extension Guide

HP UCA for Event Based Correlation - Value Pack Development Guide

HP UCA for Event Based Correlation - User Interface Guide

HP UCA for Event Based Correlation – Clustering and HA Guide

Support

Please visit our 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.

Chapter 1

New features provided by this release (V3.1)

The previous released version of UCA for EBC was UCA for Event Based Correlation V3.0.

The following features and fixes have been implemented since:

1.1 Orchestration feature

Orchestration (aka Orchestra) is a new feature introduced with UCA for EBC V3.1.

Orchestration is the recommended way of interconnecting scenarios. It makes it possible to build complex correlation solutions that span multiple scenarios.

Solution developers can use scenarios as correlation building blocks that are connected together by way of an Orchestration configuration file.

Basically, each scenario that is part of the Orchestration solution will receive alarms, process them, and hand them back to Orchestra. The Orchestration framework will decide at any point in the global correlation solution which scenario(s) to delegate/copy the alarms to next, depending on the Orchestra routing configuration file.

For full information on the Orchestration solution, please refer to the "UCA for Event Based Correlation - Reference Guide" Chapter 7.1 "Orchestration of scenarios cascading".

1.2 UCA for EBC Channel Adapter enhancements

The UCA for EBC Channel Adapter has been enhanced: all remaining JBI components have been replaced by corresponding Camel components. The UCA for EBC Channel Adapter is now fully implemented using Camel components.

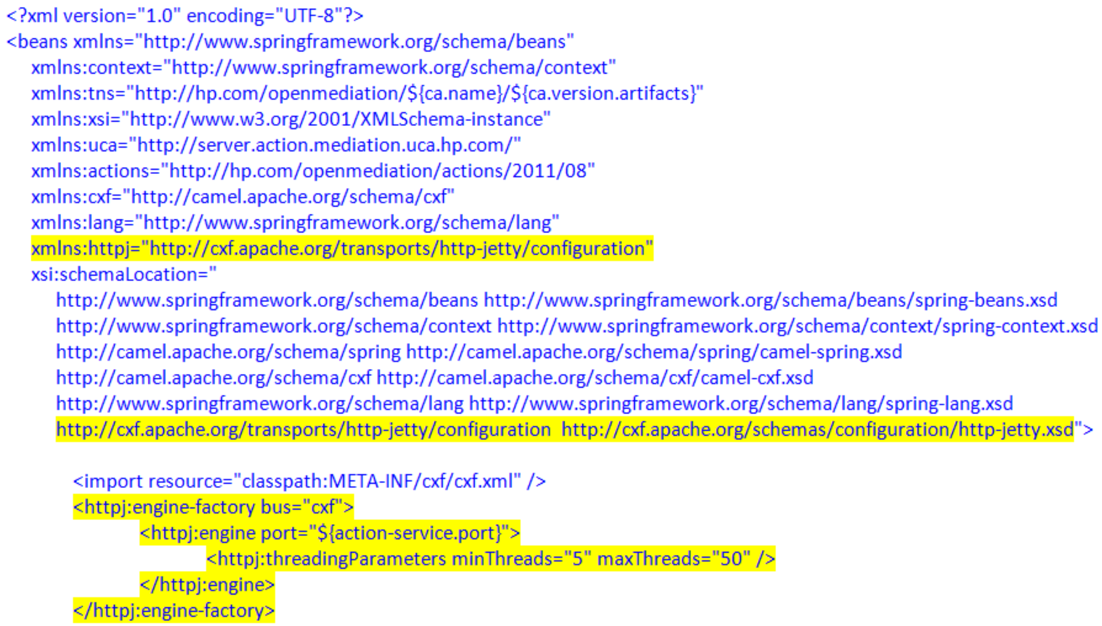
As a consequence, the procedure for setting the thread pool options of the UCA for EBC Channel Adapter has changed.

Setting the thread pool options is optional. It is recommended for production environments or if you expect the UCA for EBC Channel Adapter to process a large volume of actions.

The thread pool size options of the UCA for EBC Channel Adapter with regards to actions can be set by editing the *actions-to-nom.xml* file located in the following folder:

```
/var/opt/openmediation-70/containers/instance-<instance
            number>/ips/uca-ebc-ca-3.1/etc
```

It is recommended to add threading parameters (to the CXF component) as shown in the following screenshot (the lines to add are highlighted):

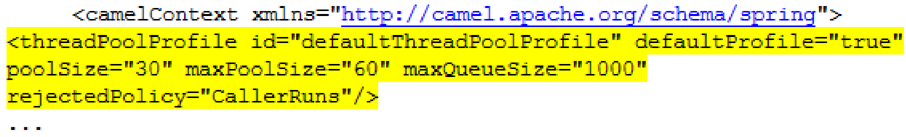


```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:context="http://www.springframework.org/schema/context"
    xmlns:tns="http://hp.com/openmediation/${ca.name}/${ca.version.artifacts}"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:uca="http://server.action.mediation.uca.hp.com/"
    xmlns:actions="http://hp.com/openmediation/actions/2011/08"
    xmlns:cxf="http://camel.apache.org/schema/cxf"
    xmlns:lang="http://www.springframework.org/schema/lang"
    xmlns:httpj="http://cxf.apache.org/transports/http-jetty/configuration"
    xsi:schemaLocation="
        http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd
        http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context.xsd
        http://camel.apache.org/schema/spring http://camel.apache.org/schema/spring/camel-spring.xsd
        http://camel.apache.org/schema/cxf http://camel.apache.org/schema/cxf/camel-cxf.xsd
        http://www.springframework.org/schema/lang http://www.springframework.org/schema/lang/spring-lang.xsd
        http://cxf.apache.org/transports/http-jetty/configuration  http://cxf.apache.org/schemas/configuration/http-jetty.xsd">

           <import resource="classpath:META-INF/cxf/cxf.xml" />
           <httpj:engine-factory bus="cxf">
                      <httpj:engine port="${action-service.port}">
                                 <httpj:threadingParameters minThreads="5" maxThreads="50" />
                      </httpj:engine>
           </httpj:engine-factory>
```

Figure 1 - Setting the CXF component thread pool size in the actions-to-nom.xml file

It is also recommended to add a thread pool profile (to the camelContext) as shown in the following screenshot (the lines to add are highlighted):



```
      <camelContext xmlns="http://camel.apache.org/schema/spring">
<threadPoolProfile id="defaultThreadPoolProfile" defaultProfile="true"
poolSize="30" maxPoolSize="60" maxQueueSize="1000"
rejectedPolicy="CallerRuns"/>
...
```

Figure 2 - Setting the Camel Context thread pool size in the actions-to-nom.xml file

1.3 The UCA for EBC GUI URL has changed

The UCA for EBC GUI is now accessible through a new URL:

`http://<host>:<port>/uca`

Where:

- *`<host>`* is a placeholder for the host name or IP address of the UCA for EBC Server instance host, *localhost* by default. This host name or IP address is defined by the *uca.ebc.serverhost* property in the *${UCA_EBC_INSTANCE}/conf/uca-ebc.properties* file
- *`<port>`* is a placeholder for the GUI port number of the UCA for EBC Server instance, *`8888`* by default. This port number is defined by the *`uca.gui.port`* property in the *`${UCA_EBC_INSTANCE}/conf/uca-ebc.properties`* file

The UCA for EBC GUI URL of previous versions of UCA for EBC (any version prior to UCA for EBC V3.1), *http://<host>:<port>/uca_expert_ui.html*, is no more supported.

1.4 UCA for EBC GUI can be made optional

It is now possible to configure UCA for EBC so that the UCA for EBC GUI web server does not start when UCA for EBC starts.

To not have the UCA for EBC GUI web server start when UCA for EBC starts, please comment the line that defines the *uca.gui.webapp* property in the *${UCA_EBC_INSTANCE}/conf/uca-ebc.properties* file by adding # at the beginning of the line as shown below:

```
…

#

# Web GUI

#

# uca.gui.webapp = webapp/uca-expert-ui.war

…
```

1.5 Value pack specific GUI extension

It is now possible to have a specific web application delivered along with a value pack.

This web application will be displayed when the value pack is started through the UCA for EBC GUI as a new tab in the value pack monitoring panel.

You can have multiple web applications for one value pack. Any *.war* file delivered with a value pack (usually in the *lib/* sub-directory) will be loaded by the UCA for EBC GUI web server and visible through the UCA for EBC GUI.

1.6 UCA for EBC GUI extras

At the UCA for EBC GUI, the “Extras” sub-menu is displayed when you have optionally put some extra *.war* files under the *${UCA_EBC_INSTANCE}/webapps* directory. Each *.war* file stored in this directory will be displayed by the UCA for EBC GUI under the new “Extras” menu, under the UCA for EBC main menu.

1.7 UCA for EBC GUI filter builder utility

At the UCA for EBC GUI, a new Filter Builder panel is available under “Filter Configuration”. It allows for a better view of existing filters displayed in a tabular mode and makes it easier to edit or create new filters. It is based on common patterns. As such, not all filters files can be displayed though this new panel.

The filters tags schema has been enhanced to support new “type” and “default” attributes.

1.8 UCA for EBC GUI upload/remove value pack features

It is now possible to upload a new value pack from your local machine (i.e. your development server) directly to a UCA for EBC server instance.

This feature is accessible from the UCA for EBC GUI, at the bottom of the page from the “Application” menu under the UCA for EBC main menu. You need developer credentials to upload/remove value packs in the UCA for EBC GUI.

It is also now possible to remove a value pack from that same panel.

1.9 Tuning a UCA for EBC Server instance JVM boot options

It is now possible to tune options used at JVM (Java Virtual Machine) startup for each UCA for EBC Server instance individually.

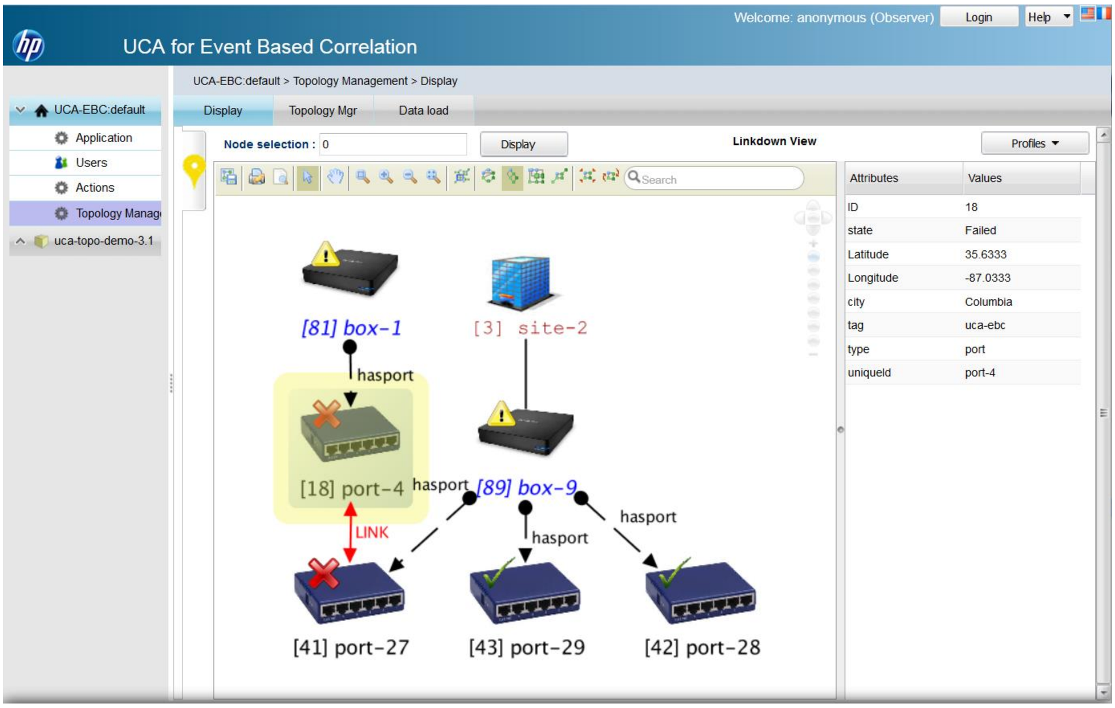
These settings are defined in the *`${UCA_EBC_INSTANCE}/conf/`***`uca-ebc.options`** configuration file.

1.10 Graphical Display of the UCA for EBC Topology

From the UCA for EBC Graphical User Interface, the operator can now display the topology graph around some Point Of Interests (POIs).

Value packs can now define POIs during the correlation process. A POI is associated to a topology node that is defined as ‘of interest’ during the correlation. The rendering of the topology graph around POIs is driven by the selection of drawing profiles that is adapted to the POI category.

Below is an example of a topology view around a Point Of Interest:



Node selection : 0

Attributes	Values
ID	18
state	Failed
Latitude	35.6333
Longitude	-87.0333
city	Columbia
tag	uca-ebc
type	port
uniqueId	port-4

Figure 3 - Example of a topology view around a Point Of Interest

For full information on this new feature, please refer to the “UCA for Event Based Correlation - Topology Extension Guide”.

1.11 GUI configuration of Orchestration feature

At the UCA for EBC GUI, there is a new sub-menu under the UCA for EBC main menu for Orchestra service configuration. With this menu, you can now configure the routes and filters of Orchestra workflows.

1.12 Performance enhancements

Some major performance enhancements are brought by UCA for EBC V3.1. The UCA for EBC Collector component in particular has been the focus of these enhancements.

1.13 Specific documentation of the TeMIP Integration

The new document named “UCA for Event Based Correlation – UCA for EBC to TeMIP Integration Guide” is now available.

This document describes in detail how to configure and tune a solution that connects UCA for EBCto TeMIP.

1.14 Data Base Alarm Forwarding

It is now possible to export alarms into a database (DB) for persistence. Alarms stored in such a DB are resynchronized at DB flow startup.

This feature is optional and works with any database that has a JDBC driver.

1.15 Data Base Alarm Flow

The new DB Alarm Forwarding feature described above provides easy DB persistence for alarms. There is a need for a value pack to know what happens to these alarms in real time.

When alarms are inserted/updated/removed from the DB, a new mechanism called “DB flow” is used to report these modifications back to the value pack.

A DB flow is optional and is configurable as part of the value pack configuration.

A DB flow supports resynchronization, automatic start, and is used at value pack startup to feed the working memory of the value pack with stored alarms.

1.16 REST API to access Alarm Data Base

The new DB Alarm Forwarding feature described above provides easy DB persistence for alarms. There is a need to retrieve those alarms from an external application (for example from HP Unified OSS Console) for display purposes or performing actions on the alarms (terminate/clear/…)

Access to the DB is provided by a REST API.

The REST API is optional. By default it is not activated.

1.17 Generic Event interface

In this version of UCA for EBC, a new generic Event interface has been introduced to support correlation of not only Alarms but also different metrics.

All Alarms Java objects have been re-factored and are now inheriting this generic Event interface.

1.18 Data Base Generic Event Forwarding

This version of UCA for EBC brings an Event Forwarding feature, like the Alarm forwarding feature already available. Any kind of generic Event can now be stored in a specific DB. However, contrary to alarms, there is no generic Event flow support.

1.19 uca-ebc-admin command-line tool enhancements

The uca-ebc-admin command-line tool has been enriched with new options:

`-a,--audit`	dumps a full audit
`-s,--stats`	displays statistics

The `-l,--list` option has been enhanced to list all deployed and not deployed value packs with their status. Before (in V3.0 and prior version), it was listing only active value packs.

Several new options for managing UCA for EBC Value Pack mediation flows have been added:

`-flow <mediation flow name>`	for performing actions on specific mediation flows
`-statusflow,--statusflow`	displays the status of a mediation flow
`-startflow,--startflow`	starts a mediation flow
`-stopflow,--stopflow`	stops a mediation flow
`-resyncflow,--resyncflow`	resynchronizes a mediation flow

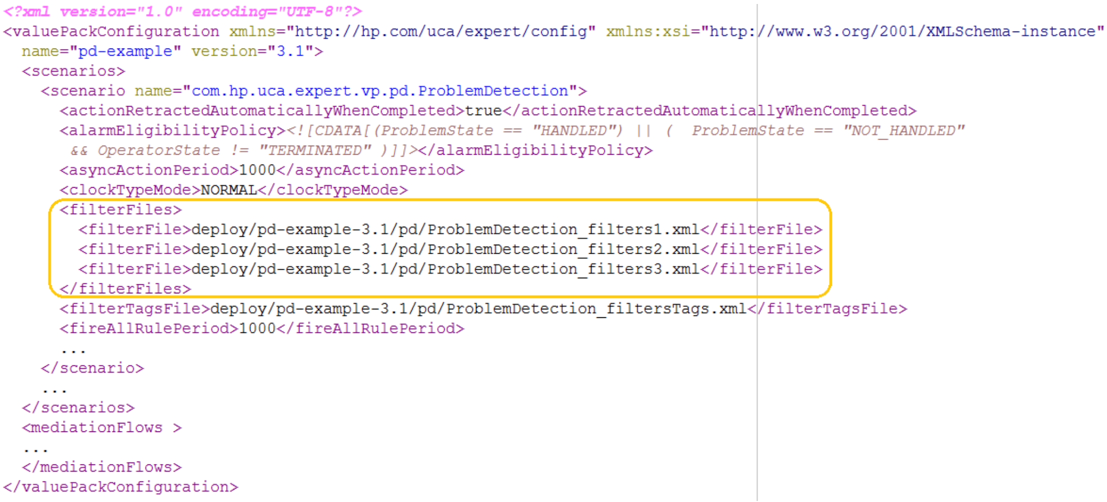
Additional options have also been added to dealing with Failed Actions:

`-dumpfa,--dumpfailedactions`	dump failed actions of a scenario to the logs.
`-retractfa,--retractfailedactions`	retract failed actions of a scenario from Working Memory

1.20 Support for multiple filter files

Support has been added for multiple filter files per scenario. It is now possible to define more than one filter file per scenario. When processing alarms for the scenario, all filter files defined for a scenario will be taken into account.

You can define multiple filter files in the ValuePackConfiguration.xml file of your Value Pack:



```xml
<?xml version="1.0" encoding="UTF-8"?>
<valuePackConfiguration xmlns="http://hp.com/uca/expert/config" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  name="pd-example" version="3.1">
  <scenarios>
    <scenario name="com.hp.uca.expert.vp.pd.ProblemDetection">
      <actionRetractedAutomaticallyWhenCompleted>true</actionRetractedAutomaticallyWhenCompleted>
      <alarmEligibilityPolicy><![CDATA[(ProblemState == "HANDLED") || (  ProblemState == "NOT_HANDLED"
       && OperatorState != "TERMINATED" )]]></alarmEligibilityPolicy>
      <asyncActionPeriod>1000</asyncActionPeriod>
      <clockTypeMode>NORMAL</clockTypeMode>
      <filterFiles>
        <filterFile>deploy/pd-example-3.1/pd/ProblemDetection_filters1.xml</filterFile>
        <filterFile>deploy/pd-example-3.1/pd/ProblemDetection_filters2.xml</filterFile>
        <filterFile>deploy/pd-example-3.1/pd/ProblemDetection_filters3.xml</filterFile>
      </filterFiles>
      <filterTagsFile>deploy/pd-example-3.1/pd/ProblemDetection_filtersTags.xml</filterTagsFile>
      <fireAllRulePeriod>1000</fireAllRulePeriod>
      ...
    </scenario>
    ...
  </scenarios>
  <mediationFlows >
  ...
  </mediationFlows>
</valuePackConfiguration>
```

Figure 4 - Referencing multiple filter files inside a ValuePackConfiguration.xml file

You can edit these multiple filter files at the GUI too.

1.21 Value pack specific GUI extension (URL service path)

It is now possible to define the Service Path of GUI extension webapps.

Please refer to “UCA for EBCValue Pack Development Guide” chapter 5.7 for details on this new feature.

1.22 Value pack specific GUI extension (URL parameters)

It is now possible to define the parameters on GUI extension webapps URL’s.

Please refer to “UCA for EBC Value Pack Development Guide” chapter 5.7 for details on this new feature.

1.23 Value pack specific GUI extension (single sign-on)

Some web application may need to take into account which UCA for EBC user is logged-on (as well as his associated role) in order to adapt its processing.

Please refer to “UCA for EBC Value Pack Development Guide” chapter 5.7 for details on this new feature.

1.24 “Impermanent” topology data base for Topology based value pack JUnit tests

It is now possible to configure the Topology database used for JUnit tests so that the data base is located in memory (instead of being on disk). This makes the test more efficient.

Please refer to the “UCA for EBC Topology Extension” guide and the topology example value pack for a full description of this feature.

1.25 Bug fixing

Please refer to chapter: Chapter 3 “Fixed Problems”.

Chapter 2

Noticeable changes compared to previous released version (V3.0)

Below is a list of noticeable changes compared to previous released version: UCA for EBC V3.0.

2.1 Changes to the AbstractJunitIntegrationTest class

The *Alarm getAlarm(String identifier)* method of the *com.hp.uca.expert.testmaterial.AbstractJunitIntegrationTest* class has been modified to throw a *com.hp.uca.expert.testmaterial.TestException* in case the method is executed in the context of a scenario in STREAM mode. Previously, this method did not throw any exception.

As a reminder, the purpose of this method is to retrieve an alarm from working memory based on its identifier, for **scenarios in CLOUD mode only**.

So if you want to migrate an existing UCA for EBC value pack to UCA for EBC V3.1 and if this value pack has JUnit tests that use the *Alarm getAlarm(String identifier)* method, then you will need to update the code of your JUnit tests to also throw (or catch) the new *com.hp.uca.expert.testmaterial.TestException* exception.

2.2 Changes to the value pack build file: build.xml

Old value pack Ant build files, *build.xml*, must be updated.

With UCA for EBC V3.1, the location of library and resource files has changed. This leads to changes in the *build.xml* file of UCA for EBC value packs.

Even if the previous *build.xml* file still works with UCA for EBC V3.1 (i.e. a value pack developed for a version of UCA for EBC prior to V3.1 should compile on UCA for EBC V3.1 with no change), it is recommended to update the *build.xml* file anyway to avoid future compatibility disruption.

Examples of new *build.xml* files that must be used can be found at the following locations:

- *${UCA_EBC_DEV}/vp-examples/pd-example/build.xml*
- *${UCA_EBC_DEV}/vp-examples/skeleton-project/build.xml*

To upgrade your value pack *build.xml* files to UCA for EBC V3.1 (assuming you did not customize them) simply replace your *build.xml* files with new UCA for

EBC V3.1 `build.xml` files (for example: `${UCA_EBC_DEV}/vp-examples/skeleton-project/build.xml`) and set the `project.name` and `project.version` properties in the new files to the name and version of your value packs.

2.3 Changes to the value pack configuration file: ValuePackConfiguration.xml

Some changes have been introduced in the value pack configuration file: `ValuePackConfiguration.xml`.

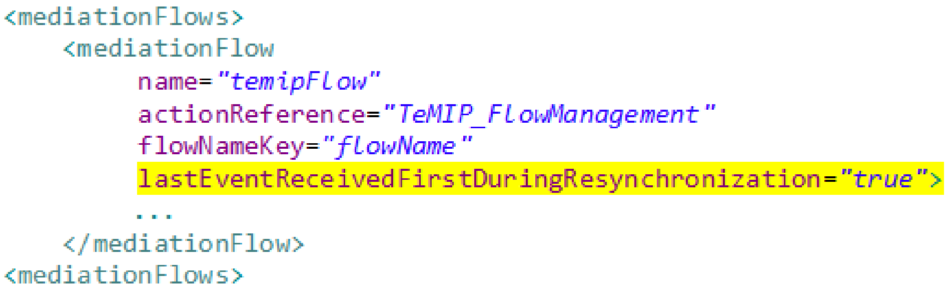
The mediation flow configuration part in this file has been enhanced in order to better manage mediation flow resynchronizations. Some alarm providers (such as TeMIP for example) are sending alarms in reverse chronological order of the alarm creation time during resynchronization, whereas some other network management systems send the alarms in chronological order during resynchronization.

To manage this possibility of a network management system to send alarms in either chronological or reverse chronological order during resynchronization, a new Boolean attribute has been added to the mediation flow definition part in the `ValuePackConfiguration.xml` file: `lastEventReceivedFirstDuringResynchronization`.

This attribute is mandatory and has no default value, so it must be added to all old (prior to V3.1) UCA for EBC value packs. Possible values for this attribute are `true` and `false`.

This attribute is mandatory and has no default value, so it must be added to all existing (prior to V3.1) UCA for EBC value packs. Possible values for this attribute are `true` and `false`.

Below is an example of how to set the new `lastEventReceivedFirstDuringResynchronization` attribute:



```
<mediationFlows>
    <mediationFlow
         name="temipFlow"
         actionReference="TeMIP_FlowManagement"
         flowNameKey="flowName"
         lastEventReceivedFirstDuringResynchronization="true">
         ...
    </mediationFlow>
<mediationFlows>
```

Figure 5 - Setting the lastEventReceivedFirstDuringResynchronization attribute in the ValuePackConfiguration.xml file

Important Note

In the current implementation, setting this attribute has no effect on non-Problem Detection value packs.

However, for Problem Detection value packs that consume alarms coming from a TeMIP system (that sends alarms in reverse chronological order of alarm creation time during resynchronization), **this attribute must be set to `true`.**

Chapter 3

Fixed Problems

This section lists the customer's visible problems that have been fixed since the last release (UCA for EBC V3.0).

Reference / Severity	Component	Description	Comment
CR#9038 Medium	Server	The "Cascading" value pack is not documented anywhere	Documented with new Orchestration solution
CR#9331 Medium	Documentation	An integration guide of UCA with TeMIP should be developed	A new document is released with the product
CR#9655 Medium	Toolkit	Target "ant all" is failing from eclipse ant plugin	Fixed with the new provided Ant version 1.9.3
CR#10025 Medium	GUI	Ability to not use UI at all	Fixed
CR#10108 Urgent	Server	Dispatcher Thread started twice	Fixed
CR#10155 Medium	Server	Uca-ebc-admin tool returns unexpected errors	Fixed
CR#10157 Medium	Toolkit	UCA-EBC eclipse plugin fails creating project for Problem Detection VP when UCA-EBC-DEVPD is not installed	Fixed
CR#10206 Medium	Topology Extension	CSV loader does not allow duplicates in secondary indexes	Fixed
CR#10224 Medium	Server	Fatal error when issuing "uca-ebc –a" without default instance	Fixed
CR#10225 Medium	Server	Allow stop and start all instances at once	Implemented
CR#10284 Medium	Server	Some javadoc.jar files are delivered under $UCA-EBC-HOME/lib directory	Fixed
CR#10366 Medium	Server	FileNotFoundException raised when XML tags for file definition are defined on two separate lines	Fixed
CR#10412 Medium	Server	DuplicateRealmException may be thrown when VP has been started with bad configuration file	Fixed
CR#10424 Medium	Server	Add a separate log file for messages/events rejected by the UCA EBC collector	Implemented
CR#10503 Urgent	Server	STREAM should not use internal HashMap (this generates a memory leak)	Fixed
CR#10513 High	Server	Wrong class loader order leads to resources badly found in jar resources instead of the "conf" directory	Fixed

Reference / Severity	Component	Description	Comment
CR#10637 Medium	Server	Combination of multiple rule files and 'disabledAtStartup=true' leads to Exception at startup	Fixed
CR#10656 Urgent	Server	Performance improvement: collector, marshaller, etc.	Fixed
CR#10657 Medium	Server	Enhance uca-ebc-admin -l,--list command to also list stopped and not deployed value packs	Implemented
CR#10757 High	Server	Possible memory leaks when group is removed from Groups container	Fixed
CR#10763 Low	Server	90 days instant on license is added even if permanent licenses are present	Fixed
CR#10815 Medium	Topology GUI	When customer network is not correctly configured, the Neo4j console fails to display in the UCA-EBC GUI	Fixed
CR#10841 Medium	Server	Parents/children fields are not updated	Fixed
CR#10848 Medium	Server	Action class should be overrideable	Implemented
CR#10898 Medium	Server	Invalid help message in install-uca-ebc-ca.sh script	Fixed
CR#10900 High	Server	UCA EBC does not start on HP-UX	Fixed
CR#11028 Medium	Server	NullPointerException calling Action.getRawText() if raw text is empty	Fixed
CR#11063 Medium	Server	Add ability to insert objects by bulk into Working Memory, firing the rules only once at the end	Implemented
CR#11115 Medium	Server	Add support for multiple filter files by scenario	Implemented
CR#11116 Medium	Server	Add support for managing the mediation flows (start/stop/resync/status) with uca-ebc-admin	Implemented
CR#11138 Medium	Server	Add support for dumping/retracting failed actions with uca-ebc-admin	Implemented
CR#11179 Medium	Server	Cascading from a scenario to another an extended alarm type (defined only in the first) throws Class Not Found Exception and makes second scenario inactive (FAILED)	Fixed
CR#11181 High	Server	UCA-EBC GUI extension: add single sign-on feature	Implemented
CR#11187 Medium	Server	Need an Alarm class copy constructor for cloning to a basic Alarm type	Implemented
CR#11237 Medium	Server	Catch rules exceptions in specific logger	Implemented
CR#11316 Medium	Server	ConcurrentModificationException in compression thread when stopping value pack	Fixed
CR#11317 Medium	Server	ConcurrentModificationException when using copyAlarmToScenarion	Fixed

Reference / Severity	Component	Description	Comment
CR#11318 Medium	Server	Error processing incoming message: ConcurrentModificationException	Fixed
CR#11320 Medium	Server	Exception while dispatching incoming RPC call due to com.google.gwt.user.server.rpc.UnexpectedException:	Fixed
CR#11452 Medium	Server	URL to access the UCA for EBC User interface is not up to date in some User Documents	Fixed

Table 2 - Fixed Problems in UCA for EBC V3.1

Chapter 4

Known Problems

This section lists problems discovered during the product test campaign and that still have to be fixed.

Reference / Severity	Component	Description	Solution/Suggested workaround

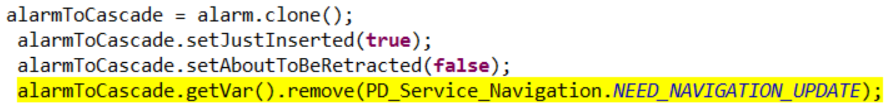
CR#9926
Medium
	Development Toolkit	Warning when building a value pack using JDK7:
		`[javac] warning: [options] bootstrap class path not set in conjunction with -source 1.6`
		`[javac] 1 warning`
			Not a real issue. Can be ignored

CR#9929 Medium	Development Toolkit	Eclipse Drools Plugin reports error in rules when creating a new VP from the skeleton using the UCA plugin. This does not prevent to run or build the VP, it looks like it is a temporary problem from Drools plugin, not impacting pure UCA-EBC	Delete the errors in the Error panel and rebuild the Value Pack project
CR#9930 Medium	Development Toolkit	Since Drools version 5.5.0-Final breakpoints in rule's code does not work anymore. This is a Drools Plugin problem	Breakpoint can be set in java Code called by the rule. Then move up in the call hierarchy to retrieve rule information.
CR#9958 Medium	Topology GUI	The Neo4J topology Management console has wrong height on Chrome.	Works fine with Chrome Release 26
CR#10000 Medium	Server	In some unclear circumstances when a value pack is stopped, some references are kept on the VP library files. This prevents to remove the files, and thus the VP can't be correctly un-deployed.	-Remove the VP directory by hand under the UCA-EBC deploy directory -This problem is not reproducible with JRE7
CR#11444 High	GUI/Server	When using the Topology graph visualisation tool from The GUI a session is created on the server side. When the browser is disconnected, the session is not fully destroyed leading to memory consumption	There is no real workaround to this problem unless stopping/re-starting the UCA-EBC server. The engineering is working on a fix that will be available in the next coming patch.
CR#11451 Medium	Server	When an alarm is copied and cascaded, the local Variables are preserved, leading to potential serious issues	See note (1) below

Table 3 - Known Problems

Notes

(1) The workaround for defect CR#11451 "*When an alarm is copied and cascaded, the local Variables are preserved, leading to potential serious issues*" is to Remove the "NeedNavigationUpdate" variable as shown below, during the cascading process:



```
alarmToCascade = alarm.clone();
 alarmToCascade.setJustInserted(true);
 alarmToCascade.setAboutToBeRetracted(false);
 alarmToCascade.getVar().remove(PD_Service_Navigation.NEED_NAVIGATION_UPDATE);
```

Figure 6 - Workaround for defect CR#11451