

Comverse ONE[®]

3.5 RT TR 3.0

Operations Reference



Notice

This document contains proprietary and confidential material of Comverse, Inc. This document is furnished under and governed by either a license or confidentiality agreement. Any unauthorized reproduction, use, or disclosure of this material, or any part thereof, is strictly prohibited.

The material furnished in this document is believed to be accurate and reliable. However, no responsibility is assumed by Comverse, Inc. for the use of this material. Comverse, Inc. reserves the right to make changes to the material at any time and without notice. This document is intended for information and operational purposes only. No part of this document shall constitute any contractual commitment by Comverse, Inc.

© 2012 Comverse, Inc. All rights reserved.

Portions of this documentation and of the software herein described are used by permission of their copyright owners.

Comverse, its logo, the spark design, and Netcentrex are registered trademarks of Comverse Technology, Inc. or its subsidiaries in the United States and may also be registered in other countries.

Other denoted product names of Comverse or other companies may be trademarks or registered trademarks of Comverse, Inc. or its subsidiaries, or their respective owners. Portions of the software may be subject to copyrights owned by Infor Global Solutions (Michigan), Inc.

Corporate Headquarters
200 Quannapowitt Parkway
Wakefield, MA 01880 USA
Tel: (781) 246-9000
Fax: (781) 224-8143
www.comverse.com

Revision History

The following table lists the document changes since the initial publication:

Date	Chapter	Description
2012/09/03	Chapter 14 - URE Process	Added a new section - System Parameters

Contents

Revision History.....	iii
Contents	v
Figures	xv
Tables.....	xvii
Notational Conventions.....	xix

Chapter 1 ASYNC Process..... 1

ASYNC Process Description	3
Prerequisites for Running ASYNC Process	3
Running ASYNC Process.....	3
Environmental Variables	3
Optimal Process Schedule	4
Reviewing ASYNC Process Activity	4
Expected Output and Location of Output	4
List of Success/Failure Criteria for ASYNC Process	4
Troubleshooting Guidelines.....	5
ASYNCH_JMS	5

Chapter 2 Voucher Recharge Process 7

Voucher Recharge Process Description	9
Prerequisites for Running Voucher Recharge Process.....	9
Running Voucher Recharge Process	9
Configuration of Voucher Recharge Process	10
Optimal Voucher Recharge Process Schedule	34
Reviewing Voucher Recharge Process Activity	34
Expected Input for Voucher Recharge Process	34
Expected Output and Location of Output	34
List of Success/Failure Criteria for Voucher Recharge Process	35
Troubleshooting Guidelines.....	35

Chapter 3 CCAP Process..... 37

CCAP Process Description	39
Prerequisites for Running CCAP Process.....	39
Dependencies on Other Processes.....	39
Running CCAP Process.....	39
Configuration of CCAP Process	39
Environmental Variables	41
System Parameters.....	41
Optimal CCAP Process Schedule	41
Reviewing CCAP Process Activity	41
Expected Input for CCAP Process	41
Expected Output and Location of Output	41
List of Success/Failure Criteria for CCAP Process	42
Troubleshooting Guidelines.....	42

Maintenance Guidelines.....	42
Chapter 4 C-MCAP Process.....	43
C-MCAP Description	45
Prerequisites for Running C-MCAP Process.....	45
Running C-MCAP Process.....	45
Configuration of C-MCAP Process.....	46
Environmental Variables	46
System Parameters.....	46
Optimal C-MCAP Process Schedule.....	48
Reviewing C-MCAP Activity	48
Expected Input for C-MCAP Process.....	48
Expected Output and Location of Output	48
List of Success/Failure Criteria for C-MCAP Process	49
Troubleshooting Guidelines.....	50
Maintenance Guidelines.....	53
Chapter 5 COM Process.....	55
COM Process Description	57
Prerequisites for Running COM Process	57
Running COM Process.....	57
Configuration of COM Process.....	58
Environmental Variables	58
System Parameters.....	58
Optimal COM Process Schedule.....	60
Reviewing COM Process Activity	60
Expected Input for COM Process.....	60
Expected Output and Location of Output	60
List of Success/Failure Criteria for COM Process	61
Troubleshooting Guidelines.....	62
Chapter 6 CSS Process.....	63
CSS Process Description	65
Prerequisites for Running CSS Process	65
Running CSS Process.....	65
Environmental Variables	65
System Parameters.....	65
Optimal CSS Process Schedule.....	66
Maintenance Guidelines.....	66
Chapter 7 MIUB Process.....	69
MIUB Process Description	71
Prerequisites for Running MIUB Process.....	71
Running MIUB Process.....	71
Configuration of MIUB Process	71
Environmental Variables	72
System Parameters.....	72

Optimal MIUB Process Schedule	72
Reviewing MIUB Process Activity	72
Expected Input for MIUB Process	73
Expected Output and Location of Output	73
List of Success/Failure Criteria for MIUB Process	73
Troubleshooting Guidelines.....	73
Maintenance Guidelines.....	74
 Chapter 8 ORLTP Process	75
ORLTP Process Description	77
Prerequisites for Running ORLTP Process.....	77
Running ORLTP Process.....	77
Configuration of ORLTP Process.....	77
Environmental Variables	77
System Parameters.....	78
Optimal ORLTP Process Schedule.....	80
Reviewing ORLTP Process Activity	80
Expected Input for ORLTP Process	81
Expected Output and Location of Output	82
List of Success/Failure Criteria for ORLTP Process	82
Troubleshooting Guidelines.....	82
Maintenance Guidelines.....	83
 Chapter 9 Product Catalog Propagation Process	85
Propagation Process Description	87
Prerequisites for Running Propagation Process	87
Running Propagation Process.....	88
Configuration of Propagation Process.....	88
Environmental Variables	89
Optimal Propagation Process Schedule.....	89
Reviewing Propagation Process Activity	89
Expected Input for Propagation Process.....	89
Expected Output and Location of Output	90
List of Success/Failure Criteria for Propagation Process.....	90
Troubleshooting Guideline	90
Maintenance Guidelines.....	91
 Chapter 10 RCS Process.....	93
RCS Process Description.....	95
Prerequisites for Running RCS Process	95
Running RCS Process	95
Configuration of RCS Process	96
Environmental Variables	96
System Parameters.....	97
Optimal RCS Process Schedule	98
Reviewing RCS Process Activity	99
Expected Input for RCS Process.....	99
List of Success/Failure Criteria for RCS Process.....	102

Troubleshooting Guidelines.....	102
Maintenance Guidelines.....	106
Chapter 11 Recharge Agent Process.....	109
Recharge Agent Process Description	111
Prerequisites for Running Recharge Agent Process.....	111
Running Recharge Agent Process.....	111
Configuration of Recharge Agent Process.....	111
System Parameters.....	112
Optimal Recharge Agent Process Schedule	113
Reviewing Recharge Agent Process Activity	113
Expected Input for Recharge Agent Process	113
Expected Output and Location of Output	113
List of Success/Failure Criteria.....	115
Troubleshooting Guidelines.....	116
Chapter 12 Reports GUI Process	119
Reports GUI Process Description	121
Installation	123
Configuration of Utility	126
Environmental Variables	130
System Parameters.....	130
Optimal Process Schedule	131
Reviewing Process Activity	131
Expected Input for Process	131
Expected Output and Location of Output	132
Troubleshooting Guidelines.....	132
Maintenance Guidelines.....	133
Chapter 13 UFA Process.....	135
UFA Process Description	137
Prerequisites for Running UFA Process.....	137
Running UFA Process.....	137
Configuration of UFA Process.....	137
Environmental Variables	137
System Parameters.....	138
Optimal UFA Process Schedule.....	138
Reviewing UFA Process Activity	138
List of Success/Failure Criteria for UFA Process	138
Troubleshooting Guidelines.....	138
Chapter 14 URE Process.....	143
URE Process Description.....	145
Prerequisites for Running URE Process	145
Running URE Process	146
Configuration of URE Process	146
System Parameters.....	149

Chapter 15 SAPI Process..... 151

SAPI Process Description	153
Running SAPI Process.....	153
Environmental Variables	153
System Parameters.....	154
Optimal SAPI Process Schedule	154
Reviewing SAPI Process Activity	154
Expected Output and Location of Output	154
List of Success/Failure Criteria for SAPI Process	154
Troubleshooting Guidelines.....	155
Back Up and Restore	156

Chapter 16 ASU Process..... 159

ASU Process Description	161
Prerequisites for Running ASU Process	161
Running ASU Process.....	161
Environmental Variables	161
System Parameters.....	161
Reviewing ASU Process Activity	162
Troubleshooting Guidelines.....	162

Chapter 17 Operational UFP Utility Details 165

UFP Utility Description	167
Prerequisites for Running UFP Utility	167
Running UFP Utility.....	167
Configuration of UFP Utility	167
Environmental Variables	168
System Parameters.....	168
HOST_CONTACTS.....	169
Optimal UFP Utility Schedule	169
Reviewing UFP Utility Activity	169
Expected Input for UFP Utility	170
Expected Output and Location of Output	170
List of Success/Failure Criteria for UFP Utility	170
Troubleshooting Guidelines.....	170
Maintenance Guidelines.....	171

Chapter 18 Operational UFR Utility Details 173

UFR Utility Description	175
Prerequisites for Running UFR Utility.....	175
Running UFR Utility.....	175
Configuration of UFR Utility	175
Environmental Variables	176
System Parameters.....	176
HOST_CONTACTS.....	177
Optimal UFR Utility Schedule.....	178
Reviewing UFR Utility Activity	178

Expected Input for UFR Utility	179
Expected Output and Location of Output	179
List of Success/Failure Criteria for UFR Utility	179
Troubleshooting Guidelines.....	179
Maintenance Guidelines.....	179
Chapter 19 CTI Functionality	181
CTI Functionality Description	183
Prerequisites for Running CTI Functionality.....	183
Environmental Variables	184
System Parameters.....	185
Reviewing CTI Functionality.....	185
Expected Input	186
List of Success/Failure Criteria for CTI Functionality	186
Troubleshooting Guidelines.....	186
Chapter 20 OCS Process	187
OCS Description.....	189
Dependencies on Other Processes.....	189
Running OCS Process	189
Configuration of OCS Process	189
Environmental Variables	189
System Parameters.....	189
Optimal OCS Process Schedule	190
Reviewing OCS Process Activity.....	190
Expected Input for OCS Process	190
Expected Output and Location of Output	190
List of Success/Failure Criteria for OCS Process.....	191
Troubleshooting Guidelines.....	191
Maintenance Guidelines.....	191
File System Maintenance	191
Database Maintenance	191
Chapter 21 OSA Process	193
OSA Description.....	195
Dependencies on Other Processes.....	195
Running OSA Process	195
Configuration of OSA Process	195
Environmental Variables	195
System Parameters.....	195
Optimal OSA Process Schedule	196
Reviewing OSA Process Activity.....	196
Expected Input Requests for OSA Process	196
Expected Output and Location of Output	196
List of Success/Failure Criteria for OSA Process.....	197
Troubleshooting Guidelines.....	197
Maintenance Guidelines.....	197
File System Maintenance	197

Database Maintenance	197
Chapter 22 USSD Process	199
USSD Description	201
Prerequisites for Running USSD Process	201
Running USSD Process	202
Configuration of USSD Process	202
Environmental Variables	202
System Parameters	202
Optimal USSD Process Schedule	203
Reviewing USSD Process Activity	203
Expected Input Requests for USSD Process	203
Expected Output and Location of Output	203
List of Success/Failure Criteria for USSD Process	203
Troubleshooting Guidelines	203
Maintenance Guidelines	203
File System Maintenance	203
Database Maintenance	204
Chapter 23 Camel 3 GPRS Process	205
Camel 3 GPRS Application Description	207
Dependencies on Other Processes	207
Running Camel 3 GPRS Process	207
Configuration of Camel 3 GPRS Process	207
Environmental Variables	208
System Parameters	208
Optimal GPRS Process Schedule	208
Reviewing GPRS Process Activity	208
Expected Input for GPRS Process	209
Expected Output and Location of Output	209
List of Success/Failure Criteria for GPRS Process	209
Troubleshooting Guidelines	209
Maintenance Guidelines	209
File System Maintenance	209
Database Maintenance	209
Chapter 24 ECI Process	211
ECI Process Description	213
Running ECI Process Utility	213
Configuration of ECI Process	213
System Parameters	213
Reviewing ECI Process Activity	213
Expected Input for ECI Process	214
Expected Output and Location of Output	215
List of Success/Failure Criteria for ECI Process	215
Troubleshooting Guidelines	215

Chapter 25 CallProcessor Process	217
CallProcessor Process Description	219
Running CallProcessor Process.....	219
Configuration of CallProcessor Process.....	219
Environmental Variables	220
System Parameters.....	220
Reviewing CallProcessor Process Activity	221
Expected Input for CallProcessor Process.....	221
Expected Output and Location of Output	221
List of Success/Failure Criteria for CallProcessor Process	221
 Chapter 26 CAMEL 3 SMS.....	 223
CAMEL 3 SMS Functionality Description	225
Running CAMEL 3 SMS Process.....	225
Configuration of CAMEL 3 SMS process	225
Environmental Variables	226
System Parameters.....	226
Reviewing CAMEL 3 SMS Functionality	226
Expected Input for CAP3SMS Process	227
Expected Output and Location of Output	227
List of Success/Failure Criteria for CAMEL 3 SMS Functionality	227
 Chapter 27 OR Manager Process	 229
OR Manager Process Description	231
Running OR Manager Process	231
Configuration of OR Manager Process	231
Reviewing OR Manager Process Activity.....	232
Expected Input for OR Manager Process.....	232
Expected Output and Location of Output	232
List of Success/Failure Criteria for OR Manager Process.....	233
Troubleshooting Guidelines.....	233
 Chapter 28 TSP Process - OMNI Mode	 235
TSP Process Description	237
Prerequisites for Running TSP Process.....	237
Running TSP Process.....	237
Configuration of TSP Process.....	237
Environmental Variables	239
Reviewing AMP Process Activity.....	239
Expected Input for TSP Process	240
Expected Output for TSP Process	240
List of Success/Failure Criteria for AMP Process.....	240
Troubleshooting Guidelines.....	240
 Chapter 29 Weblogic Admin Server - Install and Upgrade Process	

245

Weblogic Admin Server Overview 247

Install Process 249

Upgrade Process..... 265

Add Remove Scripts..... 267

Index 271

Figures

Figure 1	Product Catalog Marketing/Packaging Tab TreeView	10
Figure 2	Marketing/Packaging Tab TreeView - Right-Click Menu	10
Figure 3	Recharge Control List	11
Figure 4	Recharge Control General Tab	11
Figure 5	Recharge Control Matching Criteria Tab shows the Matching Criteria tab	12
Figure 6	Recharge Control Balance Adjustments/Awards Tab	13
Figure 7	Add Balance Window	14
Figure 8	Recharge Control Offer Adjustments/Awards Tab	15
Figure 9	New Recharge Control Offer Window	16
Figure 10	Card Generator Database Setup Screen	17
Figure 11	Configuration Window - Page One	18
Figure 12	Configuration Window – Page Two	20
Figure 13	Configuration Window - Page Three	22
Figure 14	Configuration Window - Page Four	24
Figure 15	CCBatchApplicationSettings Tags	25
Figure 16	7-Zip File Manager	26
Figure 17	Editing applicationContext.infrastructure.db.xml File	26
Figure 18	7-Zip File Manager	27
Figure 19	Editing CCBSSConfiguration.properties File	28
Figure 20	Updating CCBSSConfiguration.properties Parameters	29
Figure 21	Oracle 10g Client Installation Home Window	30
Figure 22	Installation Type Window	31
Figure 23	Specify Home Details Window	31
Figure 24	Product-Specific Prerequisite Checks Window	32
Figure 25	Summary Window	33
Figure 26	Oracle Client 10.2.0.1.0 Installation Window	33
Figure 27	Operational Reports Interface – Login Window	121
Figure 28	Operations Reports Interface	122
Figure 29	Schedule/Generate New Reports Window	122
Figure 30	View Reports Window	123
Figure 31	Report Formats	123
Figure 32	Welcome Window	124
Figure 33	Browse Install Directory	124
Figure 34	Installing Comverse ONE ORI	125
Figure 35	Installation Complete	125
Figure 36	Open Administrative Tool	126
Figure 37	System DNS Tab	126
Figure 38	Create New Data Source	127
Figure 39	Microsoft Access Driver	127
Figure 40	Select Database	128
Figure 41	Set Advanced Options	128
Figure 42	Report Status Data Loader Process State Transitions	130
Figure 43	Report Output	132
Figure 44	SAPI Admin Console Window	156
Figure 45	Example - Monitor Page	163
Figure 46	Menu Item - Indicate Ready or Busy	184
Figure 47	Architecture	248

Tables

Table 1	Notational Conventions.....	xvii
Table 2	Labels in Markers	xviii
Table 3	Types of Markers	xviii
Table 4	System Parameters	72
Table 5	System Parameter	78
Table 6	Output and Location of Output.....	82
Table 7	RC_TERM_INST/NRC_TERM_INST Possible Processing Status	96
Table 8	System Parameters	97
Table 9	Error Codes	104
Table 10	Table Structure	112
Table 11	Error Code and Error Description	113
Table 12	Reports	131
Table 13	XML files and XSD files	147
Table 14	System Parameters	149
Table 15	System Parameters	168
Table 16	System Parameters	176
Table 17	System Parameters	185

Notational Conventions



Useful information appears in this format.



Provides direction to important information



Important information appears in this format.



Indicates possible risk of damage to data, software, or hardware.



Indicates serious risk of damage to data, software, or hardware.

Table 1 Notational Conventions

Notation	Explanation of Convention
<i>References to printed documents</i>	<i>Helvetica italic</i> Example: See <i>Database Reference Volume 2</i> .
<KEYS>	UPPERCASE HELVETICA, in angle brackets Example: Press <CTRL><Q><SHIFT><P> to create an em dash.
User-entered text	Courier bold Example: Enter Total Charges in the field.
<i>Placeholders for user-determined text</i>	<i>Courier italic</i> , in angle brackets Example: Enter your <password>.
Code samples, TABLE_NAMES, field_names, file and directory names, file contents, user names, passwords, UNIX ENVIRONMENT_VARIABLES	Courier
<i>Placeholders for system-generated text</i>	<i>Helvetica italic</i> Example: Messages appear in this form: <i>timestamp messageId >> text</i> .
Buttons, Icon Names, and Menu items	Helvetica bold Example: Choose Reports from the main menu.

Special Markers

The Comverse ONE Billing and Active Customer Management solution has the three derivatives shown in [Table 2, “Labels in Markers.”](#) For user convenience, any content that is specifically included in a derivative is highlighted with special markers so that it can readily be distinguished.

Table 2 Labels in Markers

Derivative	Label Shown in Markers
Comverse ONE Converged Billing derivative	Converged only
Comverse ONE Real-Time Charging derivative	Real Time only
Comverse ONE Postpaid Billing derivative	Postpaid only

Each derivative has a set of three color-coded markers, as shown in [Table 3, “Types of Markers.”](#) The markers are used individually or in combination to highlight derivative-specific content by:

- Entire chapters
- Selected portions of chapters
- Tables, either entire or partial

Table 3 Types of Markers

Marker	Example	Description
Alert		<ul style="list-style-type: none"> ■ Placed at the beginning of an entire chapter that pertains only to a specific derivative. ■ Placed just before a table that partially or entirely pertains only to a specific derivative.
Block		A shaded box that encloses sections of documentation that pertain only to a specific derivative.
Flag		<ul style="list-style-type: none"> ■ Designates a shaded table row whose contents pertain only to a specific derivative. ■ In a bulleted list, designates an item that pertains only to a specific derivative.

Additional Documentation

For a list of other Comverse ONE documents, see the *Comverse ONE Documentation List*.

Chapter 1

ASync Process

1

orded (sender
The desti
notifying
ng The noti
ieve The m
cT access To

e

ASYNC Process Description

ASYNC/Batch Server is a back-end utility application used to process Single API calls asynchronously or in bulk.

Prerequisites for Running ASYNC Process

Dependencies on Other Processes

It is not dependent on other Processes.

Running ASYNC Process

From UPM

The version 3.0 does not support but the version 3.1 supports it.

Using mshell:

To start ASYNC/Batch server, run the following command:

```
execute_process -g application -p async -t NI -d start
```

To stop ASYNC /Batch server, run the following command:

```
execute_process -g application -p async -t NI -d stop
```

From Command Line

To start ASYNC/Batch server, perform the following steps:

1. Log in as cbsuser.
2. Go to ASYNC /Batch server installation directory.
3. Run startBatch script.

Example:

```
cd /home/cbsinstall/asynch_batch/batch
./startBatch.sh
```

To stop ASYNC/Batch server, perform the following steps:

1. Log in as cbsuser.
2. Go to ASYNC /Batch server installation directory.
3. Run shutdownBatch script.

Example:

```
cd /home/cbsinstall/asynch_batch/batch
./shutdownBatch.sh
```

Environmental Variables

The Environmental Variables that control the ASYNC's behavior are:

- Need login - cbsuser
- JAVA_HOME
- WL_HOME

Optimal Process Schedule

This process can be run whenever it is required.

Reviewing ASYNC Process Activity

For 3.0, the log is under <Async/Batch_installation>/log.

Example: /home/cbsinstall/asynch_batch//log/SingleApi.log

For 3.1, the log is under <Async/Batch_installation>/logs.

Example: /home/cbsinstall/asynch_batch/batch/logs/batch_async.log

Expected Output and Location of Output

After starting ASYNC/Batch, it gives the following output if it is successful:

```
> ./startBatch.sh
PID=17240
STATUS=0
INFO=Batch Started
```

After stopping ASYNC/Batch, it gives the following output if it is successful:

```
> ./shutdownBatch.sh
STATUS=0
INFO=Done
```

List of Success/Failure Criteria for ASYNC Process

If ASYNC/Batch runs successfully, then it gives the output as mentioned in the Expected Output and Location of Output section.

Check if the batch process is running or not using the following command:

```
ps -ef | grep java
```

If start/stop script fails to run successfully, it gives the following information:

Example 1:

```
> ./startBatch.sh
Error: Current user is not cbsuser
/app/bea/weblogic_10.3/wlserver_10.3/server/lib/wlfullclient-
10.3.jar: No such file or directory
Didn't get a valid wlfullclient-10.3.jar under /app/bea/weblogic_
10.3/wlserver_10.3/server/lib
Please check if WL_HOME is set correctly...
```

Example 2:

```
-bash-3.00$ ./shutdownBatch.sh  
Error: Current user is not cbsuser
```

Troubleshooting Guidelines

Check if the ASYNC/Batch is running or not using the following command:

```
ps -ef | grep java
```

ASYNCH_JMS

Failed Managed Server must be manually removed from the configuration to prevent future assignment to the failed Managed Server.

If any of the managed Nodes goes down, then you have to manually configure the managed nodes and restart all available managed Nodes.

You need to configure the
/home/asynclms/server/domain/config/CCBSConfiguration.properties for the all available server with the following entry.

Examples:

When three initial JMS servers are configured (original configuration) the subscr_no is divided by 3 and based on the remainder (0, 1, or 2), the ASYNCH_REQUEST is assigned to JMS servers which are configured for 0, 1, or 2.

Original Configuration:

```
async.jms.loader.count = 3 (Number of available JMS server)  
async.jms.loader.MS_kdpm228 = 0  
async.jms.loader.MS_kdpm235 = 1  
async.jms.loader.MS_kdpm236 = 2
```

If kdpm228 server is down and the following configuration is required, then the subscr_no is divided by 2 and based on remainder (0 or 1), the ASYNCH_REQUEST is assigned to JMS servers which are configured for 0 and 1 only.

Updated Configuration after failure of one managed Node:

```
async.jms.loader.count = 2  
async.jms.loader.MS_kdpm228 = 2  
async.jms.loader.MS_kdpm235 = 0  
async.jms.loader.MS_kdpm236 = 1
```

Back Up and Restore

Backing Up Profile

Use backupConfig.sh to take a backup of the Profiles that are created using WebSphere.

Location of the file: Installed Directory/Domain Manager/bin/

Example: /data/asynclms/DMGR/bin/backupConfig.sh

By default the backup stores the content in the same location. To specify a different location, change the target path in `asyncjms-backup.cfg` (`chk.script.path = ${cbs.asyncjms.server.installdir}/bin/backup.sh`) which is available in the Installed directory/server/config folder.

Example: `/data/asyncjms/server/config/asyncjms-backup.cfg`

Restoring Profile

Use `restoreConfig.sh` to restore the Profile from the backup Profile.

Location of the file: Installed Directory/Domain Manager/bin/

Example: `/data/asyncjms/DMGR/bin/restoreConfig.sh`

Backing Up Deployment Manager and Managed Servers

To back up Deployment Manager and Managed Servers for WebSphere, perform the following steps:

1. Run the backup job from UPM or run `backup.sh`.
`/home/asyncjms_ws/server/bin - ASYNCJMS`
 A zip file (`/var/msf/transfers/backup/cbs_<component>_backup-${hostName}.zip`) is created.
2. Transfer the zip file to the UPM for storage.

Restoring Deployment Manager and Managed Servers

To restore Deployment Manager and Managed Servers for WebSphere, perform the following steps:

1. Transfer the backup file to the host machine and save it under the `root` directory (`/`).
2. Run `unzip o cbs_<component>_backup-${hostName}.zip`
`cd /home/<component>_ws`
 Change Ownership
`chown -R cbsuser:users * ===== for ASYNCJMS`



NOTE

While restoring Deployment Manager or Managed Server, the IP and the hostname must be the same as it was before. When a new box is used, the setup for restoring it must have the same IP and hostname. This is necessary for the network setup. The IP/hostname is already configured for the application.

3. Once the restoring process is complete, start the servers.

Chapter 2

Voucher Recharge Process

2

orded (sender
The desTir
noTiEying
ng The noTi
ieve The m
cT access To

e

v

Voucher Recharge Process Description

Recharging can be accomplished in several ways; most commonly through recharge vouchers.

Comverse ONE solution supports the creation of recharge vouchers, generation of unique PINs, maintenance and consumption of vouchers.

Voucher-based recharging of a prepaid balance is a simple process of contacting the Comverse ONE solution (via IVR, USSD, Customer Care, Comverse ONE SV Self-Service, or other access methods) and entering the voucher number printed on the voucher. The Comverse ONE solution verifies the identity of the subscriber and the validity of the voucher, and then recharges the prepaid balances and marks the voucher as used to prevent fraudulent re-use.

Prerequisites for Running Voucher Recharge Process

Recharge vouchers alone can only recharge the core balance; the Recharge Control Table can be configured to allow replenishment of any, or all, prepaid balances, of any unit type, by setting predefined amounts in the unit types for each balance. The Recharge Control Table is provisioned in the Product Catalog.

Like subscriber records, recharge vouchers must be created, deleted, viewed, and modified. The Comverse ONE solution provides Card Generator software for creating batches of encryption protected recharge vouchers. The voucher generation process produces a file that can be delivered to the voucher manufacturer, and a file that can be read into the Comverse ONE solution voucher database, to enable tracking of voucher status through the voucher life cycle.

Creating new recharge vouchers on the Comverse ONE solution Card Generator is only a part of the voucher management process. The vouchers must also be physically printed and distributed so that the subscribers can buy them and use them to recharge their accounts.

Additional voucher-related tasks must typically be performed by customer service representatives, including managing the voucher inventory, looking up specific vouchers, and modifying individual vouchers or groups of vouchers. For example, recharge vouchers must be activated (put in the Active state) before subscribers can use them.

Running Voucher Recharge Process

Comverse ONE solution supports the traditional method of recharging by voucher. The subscriber purchases a voucher and applies the credit to their account via the recharge process. The subscriber can make the recharge request via the following recharge methods:

- **IVR:** Subscriber calls the IVR recharge menu (either directly to a dedicated number for recharging or by selecting the recharge option from the IVR menu) and enters the voucher number via DTMF input. When the recharge succeeds, the IVR announces the new balances and expiration dates.
- **USSD:** Subscriber sends a specific USSD request voucher number in the USSD string. The USSD response back informs the subscriber about the recharge result, including the updated balances and the new expiration date. USSD services are only available in certain kinds of wireless networks.
- **Comverse ONE SV Self-Service:** Subscriber makes the recharge request via the web. The subscriber enters the voucher number and after successful recharge the system displays the updated balances and expiration dates.
- **Customer Care:** Subscriber calls the customer care center and provides the voucher number. The CSR performs the recharge operation for the subscriber.

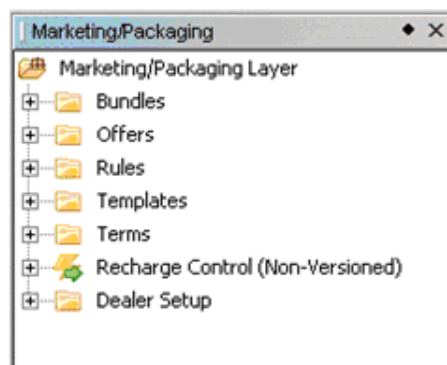
Configuration of Voucher Recharge Process

Configuring Recharge Control Table

 **To configure Recharge Control Table:**

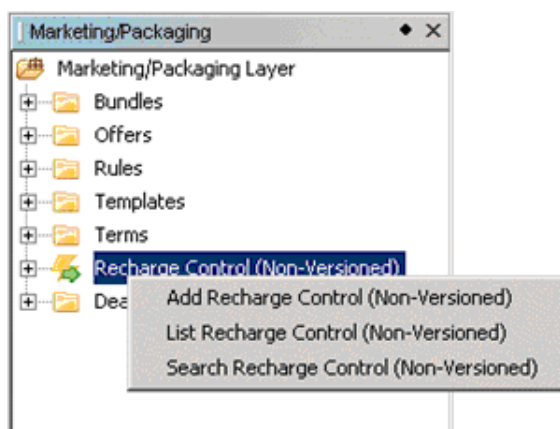
1. Click the **Marketing/Packaging** tab in the Product Catalog interface. If the tab is not already displayed, select **Marketing/Packaging** from the Window menu.
The **Marketing/Packaging** TreeView is displayed below.

Figure 1 Product Catalog Marketing/Packaging Tab TreeView



2. In the **Marketing/Packaging** TreeView, right-click the **Recharge Control (Non-Versioned)** folder and select **List Recharge Control (Non-Versioned)** in the options provided.

Figure 2 Marketing/Packaging Tab TreeView - Right-Click Menu



The Recharge Control List is displayed as shown below.

Figure 3 Recharge Control List

Priority	Name	Start Date	End Date	Low Face Value	High Face Value	Unit Type	Batch Number	Offer	Reseller	Application	Currency
1	ret1	2008-06-11 00:00:00...	2008-06-13 00:00:00...	1	3	Money/Currency	-1	-1	Global Reseller	7	US Dollars
2	ret2			1	3	Money/Currency	-1	-1	Global Reseller	7	US Dollars
3	ret3	2008-06-23 00:00:00...		1	2	Money/Currency	-1	-1	Global Reseller	7	US Dollars
4	ret4			2	5	Money/Currency	2	-1	Global Reseller	7	US Dollars
5	krsta ret			0	5	Money/Currency	-1	-1	Global Reseller	7	US Dollars
6	ret5	2008-06-27	2008-07-27	10	100	Money/Currency	400	-1	Global Reseller	Unknown propert...	US Dollars

3. Click **Edit** at the bottom of the list window to edit an existing list entry or click **Add** to add a new list entry.

A **Search** field is also provided to search a specific entry.

Adding a New Recharge Control Entry

1. When adding a new recharge entry or selecting an existing entry, the editable fields become available on the Recharge Control window.

Figure 4 Recharge Control General Tab

Recharge Control (Non-Versioned) Version 1.0

Name*

Description

Short Display

Language
English Add/Remove

General | Matching Criteria | Balance Adjustments / Awards | Offer Adjustments / Awards

Priority
1

The top area of the window displays the following fields:

- ☐ Name
- ☐ Description
- ☐ Short Display

General Tab

The General tab displays only the Priority field. When a new entry is added, it is added to the bottom of the Recharge Control List and the Priority field cannot be modified. After the entry is added to modify the priority, select the entry and click **Edit**.

Matching Criteria Tab

Figure 5 Recharge Control Matching Criteria Tab shows the Matching Criteria tab

The screenshot shows a software window titled "New RechargeControlKey * - Recharge Control (Non-Versioned) Version 1.0". The "Matching Criteria" tab is selected. The form contains the following fields:

- Name***: Text input field.
- Description**: Text input field.
- Short Display**: Text input field.
- Language**: Dropdown menu with "English" selected and an "Add/Remove" button.
- Recharge Effective Dates**:
 - Start Date**: Date picker.
 - End Date**: Date picker.
- Face Value**:
 - Units***: Dropdown menu with "Money/Currency" selected.
 - Currency***: Dropdown menu.
 - Low Face Value***: Text input field.
 - High Face Value***: Text input field.
- Offer**:
 - Primary Offer**: Dropdown menu.
- Miscellaneous**:
 - Recharge Channel***: Dropdown menu.
 - Batch Number**: Text input field.
 - Recharging Entity Type**: Dropdown menu with "Subscriber" selected.

The Matching Criteria tab displays the following fields:

Recharge Effective Dates

- ❑ **Start Date:** Start date of the recharge event. The start date is in the format - yyyy/mm/dd.
- ❑ **End Date:** End date of the recharge event. The end date is in the format - yyyy/mm/dd.

Face Value

- ❑ **Units:** Units of the recharge. Lists all the units provisioned on the system. The units selected must be defined in the system. The default unit is Currency. The voucher recharge only supports Currency.
- ❑ **Currency:** Currency of the recharged entity. If a Primary Offer is specified, this field must match the currency of the Primary Offer. Otherwise, it can be any currency defined in the system, or ANY.
- ❑ **Low Face Value:** Low monetary value of the original recharge. The ranges are exclusive. Therefore the Low Face Value must be less than the High Face Value (not greater than or equal to). These fields are never blank.
- ❑ **High Face Value:** High monetary value of the original recharge. The ranges are exclusive. Therefore the High Value must be greater than the Low Value (not greater than or equal to). These fields are never blank.

Offer

- **Primary Offer:** Primary offer of the subscriber. The choices available from the dropdown menu include ANY and the names of each primary offer provisioned for the selected reseller. Only the primary offers associated with the current reseller are displayed in the primary offer field. If ANY is selected, all the primary offers, for the selected reseller, are available.

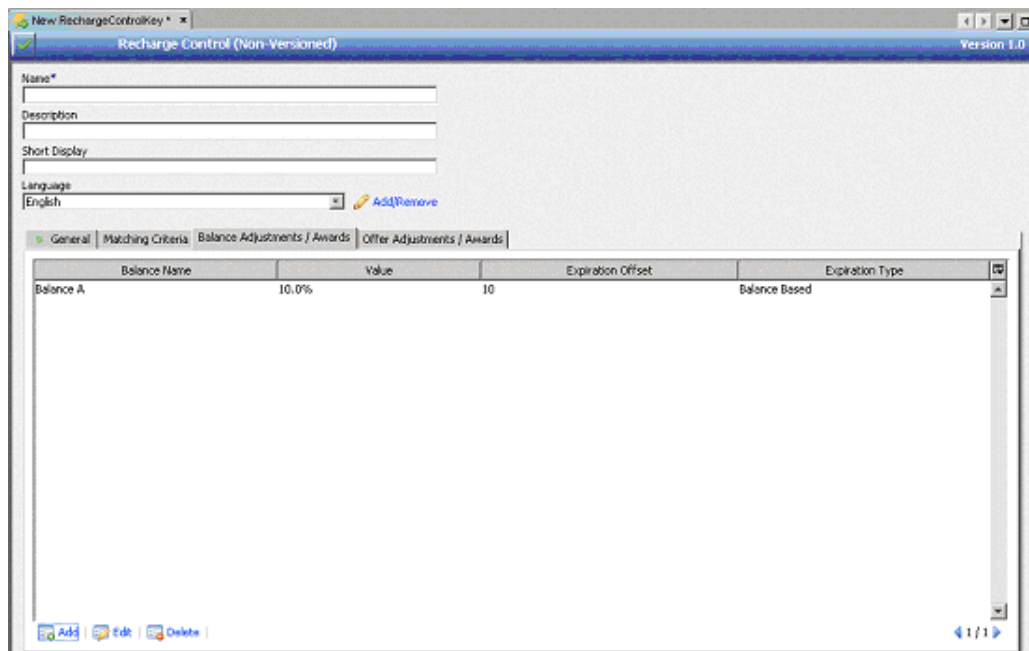
Miscellaneous

- **Recharge Channel:** Recharge method. The following options are available:
 ANY
 External_non_voucher
 External_voucher
 Fast_IVR_Voucher
 IVR_Vouchers
 Reserved for Future Internal Purpose
 Roaming
 Self_Care_voucher
 Self_Care_non_voucher
 UNKNOWN
 USSD_Voucher
- **Batch Number:** Specific batch number. If this field is left blank, the batch number is not considered and any batch number produces a match.
- **Recharging Entity Type:** Subscriber or Account.

Balance Adjustments/Awards Tab

The Balance Adjustments/Awards tab displays the current recharge details for the primary offer balances.

Figure 6 Recharge Control Balance Adjustments/Awards Tab



- ❑ **Balance:** Name of the balance that is affected by the recharge process under this offer.
- ❑ **Value:** The amount to be added to or subtracted from this balance when a recharge is performed.
- ❑ **Expiration Offset:** Additional expiration offset days added to the expiration offset face value of the recharge and has to be included in the core balance expiration date calculation. If a value is specified then the total expiration offset is the amount originally specified plus the core Balance Amount. By default, this field is blank and additional days are not added.
- ❑ **Expiration Type:** Type of expiration. Only balance-based expiration is supported, it means that the expiration for the granted amount is tied to the balance expiration date.



To add a balance to the Recharge Control Table:

1. Select the Balance Adjustments/Awards tab. Click **Add**. The Add Balance window is displayed.

Figure 7 Add Balance Window

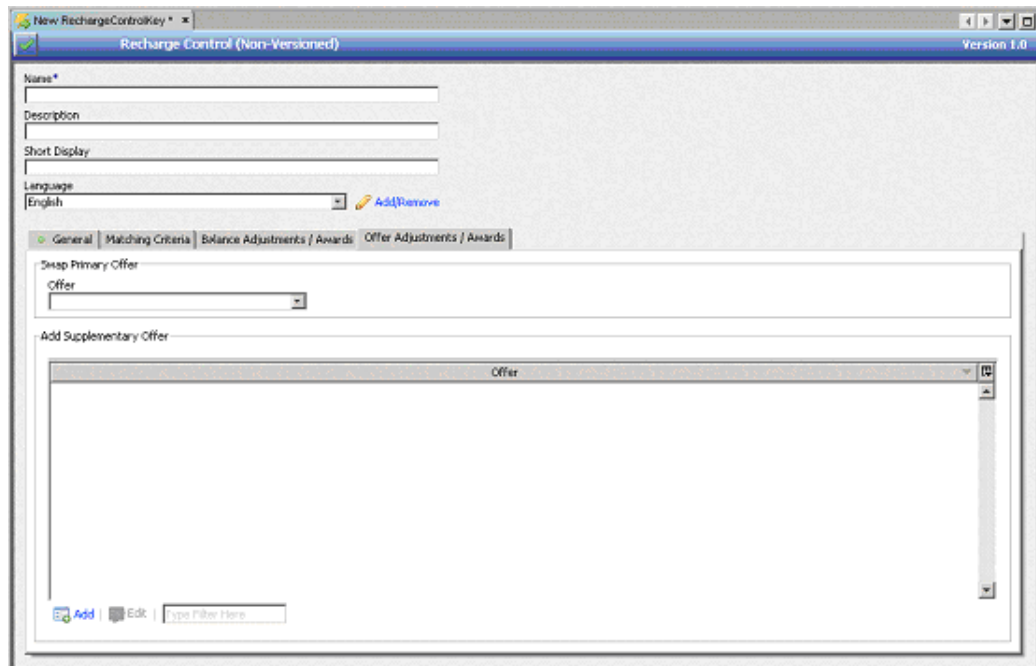
- **Balance**
The **Name** dropdown list displays all the balances associated with the current offer. Select a balance to be added from the list to associate it with the new Recharge Control instance.
The **Unit Type** field displays the unit type associated with this balance, but cannot be modified from this window.
The **Is Core** checkbox is checked if the balance is the core balance for the subscriber or account. This checkbox cannot be modified by the user.
- In the **Value** field, enter a value to be added or subtracted from the face value of the voucher.
- If the **Use a Percentage** checkbox is checked, the Percentage field is enabled and the Value field is disabled
- In the **Percentage field** enter a value to indicate the percentage of the face value of the voucher to be added to this balance.

- In the **Expiration Offset** field, enter the number of days to be added to the expiration date of the balance on a successful recharge.
- 2. Click **Save**.
- 3. Repeat these steps to associate additional balances with this Recharge Control instance.

Offer Adjustments/Awards Tab

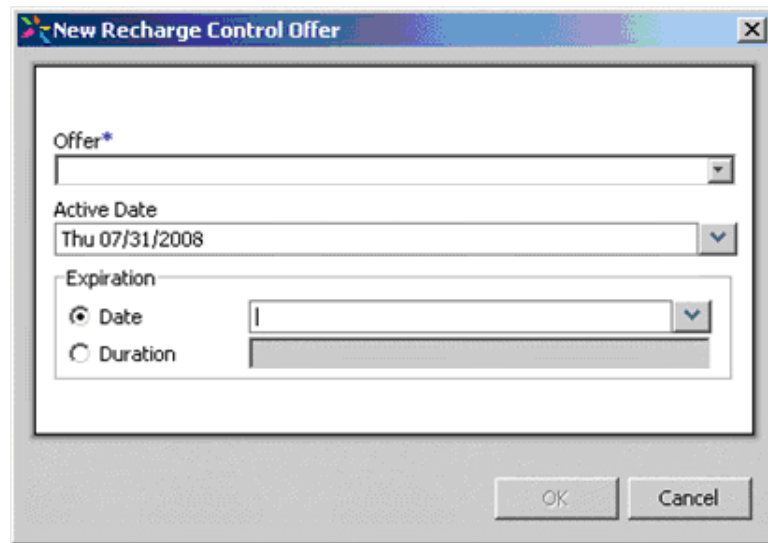
The Offer Adjustments/Awards tab displays the primary and supplementary offers currently associated with a specific RCT row.

Figure 8 Recharge Control Offer Adjustments/Awards Tab



- **Swap Primary Offer:** Specifies that this offer replaces the primary offer on a successful recharge.
- **Add Supplementary Offer:** Specifies that this offer is added to the subscriber or the account on a successful recharge.

The New Recharge Control Offer window enables adding a new supplementary offer to the Recharge Control Table.

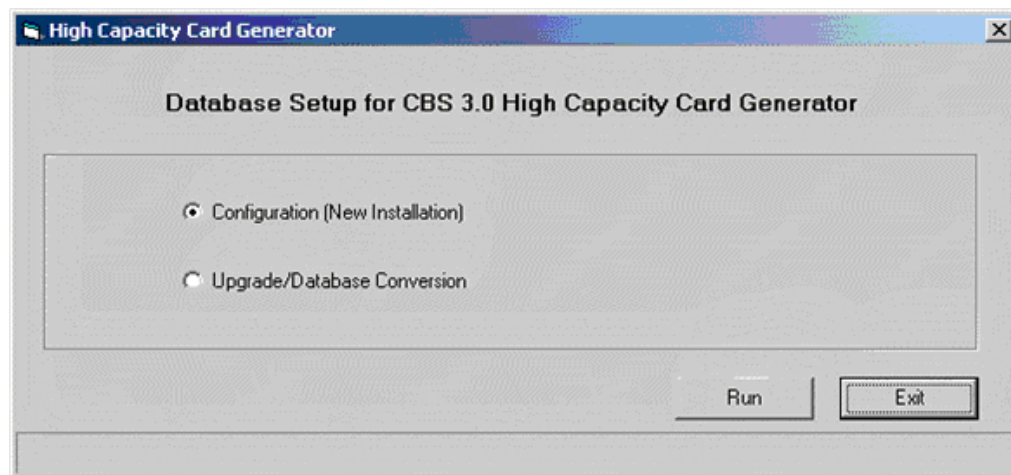
Figure 9 New Recharge Control Offer Window

- **Offer:** Select an offer from the dropdown list.
- **Active Date:** The date on which the offer becomes active.
- **Expiration:** Two options are available for setting the expiration date of the offer:
 - **Date:** Select to set an explicit expiration date. When this button is selected, a calendar appears that enables the user to select a date.
 - **Duration:** Select to set the number of days during which this offer is active and is available to the user.

Configuring Card Generator

To configure Card Generator:

1. Once the Card Generator is installed, the **Database Setup** window is displayed.

Figure 10 Card Generator Database Setup Screen

2. For a new installation, select **Configuration (New Installation)** and click **Run**.

The Configuration window ([Figure 11, "Configuration Window - Page One"](#)) is displayed. There are four pages that must be completed in the Configuration window.

The values entered during the configuration process are the default values that are used by the Card Generator when creating a batch of vouchers. These values can be modified later when the **System Options** is selected from the Administrative menu, or, for individual batches, on the page that appears when a new batch is created.



To enter values on Page One of Configuration window:



NOTE

Reseller, primary offer, bundle, and language options are defined on the first page of the Configuration window.

Figure 11 Configuration Window - Page One

The screenshot shows a 'Configuration' window with the following sections:

- Reseller:** A list box containing 'Global Reseller'. To its right are 'Name' and 'ID' input fields, and '<< Add' and 'Remove >>' buttons.
- Primary Offer:** A list box (empty). To its right are 'Name' (containing 'New Offer') and 'ID' (containing '001') input fields, and a 'Core Balance Grant Amount' input field. Below these are '<< Add' and 'Remove >>' buttons.
- Bundle:** A list box (empty). To its right are 'Name' and 'ID' input fields, and '<< Add' and 'Remove >>' buttons.
- Language:** A list box (empty). To its right are 'Name' and 'ID' input fields, and '<< Add' and 'Remove >>' buttons.

At the bottom of the window are five buttons: '<< Previous', 'Next >>', 'Finish', 'Cancel', and 'Help'.

1. In the Reseller frame, create a list of valid resellers to display in the selection list when creating voucher batches.

To add a reseller to the list:

- a. In the **Name** field, enter the name of the reseller.
- b. In the **ID** field, enter the reseller's ID.
- c. Click **<< Add**.

To remove a reseller from the list:

- a. From the list of resellers, select the reseller.
- b. Click **Remove >>**.

2. In the Primary Offer frame, create a list of valid primary offers to display in a selection list when creating voucher batches.

To add a primary offer to the list:

- a. In the **Name** field, enter the name of the primary offer.
- b. In the **ID** field, enter the primary offer ID.
- c. In the **Core Balance Grant Amount** field, enter a currency amount that is allocated to the core balance on each successful recharge.
- d. Click **<< Add**.

To remove a primary offer from the list:

- a. From the list of primary offers, select the primary offer.
- b. Click **Remove >>**.

3. In the Bundle frame, create a list of valid bundles to display in a selection list when creating voucher batches.

To add a bundle to the list:

- a. In the **Name** field, enter the name of the bundle.
- b. In the **ID** field, enter the bundle ID.
- c. Click << **Add**.

To remove a bundle from the list:

- a. From the list of bundles, select the bundle.
- b. Click Remove >>.

4. In the Language frame, create a list of valid languages to display in a selection list when creating voucher batches.

To add a language to the list:

- a. In the **Name** field, enter the name of the language.
- b. In the **ID** field, enter the language ID.
- c. Click << **Add**.

To remove a language from the list:

- a. From the list of languages, select the language.
- b. Click Remove >>.

5. Click **Next>>** to move to the next page of the Configuration window.



To enter values on Page Two of Configuration window:



NOTE

Various system options and default values used in the creation of new batches are defined on the second page of the Configuration window.

Figure 12 Configuration Window – Page Two

Configuration

System Options

Purge Events: ☐ Never ☒ After 3 ☒ Months ☐ Years

Minimum Password Length: ☐ _____

"Print House" Export File Encryption: ☐ _____

Card Code Hashing: ☐ _____

Fractional Face Value: ☐ _____

Default Export Directory: F:\

Batch Details

Pseudo-Random Code Length(9-30): 12

Suggested Initial Batch: 10000

Suggested Batch Size: 1000

Suggested Face Value: 150

Suggested Expiration Period Offset(days): 0

Suggested Serial Number: 10000

Suggested Expiration Period (months): 12

<< Previous Next >> Finish Cancel Help

1. From the **Purge Events**: radio buttons, select **After** or **Never**. If you select **After**, enter a number in the field provided, and select either the **Months** or the **Years** radio button.
2. Select the **Minimum Password Length** checkbox to set a minimum length for the password, and then enter the number of characters in the **Characters** field.
3. Select the **"Print House" Export File Encryption** checkbox to encrypt the Card Generator output files. The print house uses the decryption tool and a password to decrypt the output files.
4. Select the **Card Code Hashing** checkbox to store recharge voucher numbers in the database in a hashed format. This format uses a hashing algorithm that works in conjunction with a five-digit alphanumeric key, entered in the field beside the checkbox. When the checkbox is not selected, the key field is not accessible, and no hashing is used.
5. Select the **Fractional Face Value** checkbox to enable generation of vouchers with fractional currency values.
6. In the **Default Export Directory** field, enter the designation of the device that is most often used to capture the card batches.
7. In the **Pseudo-Random Code Length (9-30)** field, enter the length of the code number used to the activate recharge cards. It can be 9 to 30 characters long. It cannot be modified after the initial configuration.
8. In the **Suggested Initial Batch** field, enter an identifying number for the initial batch. This starting batch number is automatically increased by one each time a batch is created. It can also be modified manually when creating a new batch.
9. In the **Suggested Batch Size** field, enter a default value for the batch size. This field can be modified during the batch creation.

10. In the **Suggested Face Value** field, enter a default face value. Any positive number (including fractional values) or zero can be used. This field can be changed during the batch creation.
11. In the **Suggested Expiration Period Offset (days)** field, enter the default number of days a subscriber or account is extended when a voucher from this batch is used. This can be a positive integer or zero. This field can be changed during the batch creation.
12. In the **Suggested Serial Number** field, enter a default serial number for the first voucher in the batch. Any number can be used. This field can be changed during the batch creation.
13. In the **Suggested Card Expiration Period (months)** field, enter the default number of months after which the vouchers expire. During batch creation, this period is translated into a date. This field can be changed during the batch creation.
14. Click **Next>>** to move to the next page of the Configuration window.



To enter values on Page Three of Configuration window:



NOTE

Default values related to how batches are handled are defined on the third page of the Configuration window.

Figure 13 Configuration Window - Page Three

1. The **Possible Batch Statuses** frame contains a field, which has a default list of possible statuses provided with the system: New, In Print, Distributed, and Expired.

To add a new status to the Possible Batch Statuses list:

- a. In the **New Status** field, enter the name of the status.
- b. Click **<< Add**.

To remove a status from the Possible Batch Statuses list:

- a. Select the Status from the list of statuses.
- b. Click **Remove >>**.

From the **Default Status** dropdown list, select a default status to use during batch creation.

2. The **Currency Unit** frame contains field, which lists the currencies supported by Comverse ONE solution.

To add a new currency unit to the Currency Unit selection list:

- a. In the **Name** field, enter the name of the currency unit.
- b. In the **ID** field, enter an ID for the new currency unit.
- c. Click **<< Add**.

To remove a currency unit from the Currency Unit selection list:

- a. Select the unit from the **Currency Unit** selection list.
- b. Click **Remove >>**.

3. In the Account Segment frame, create a list of Account Segment to display in a selection list.

To add a new Account Segment to the selection list:

- a. In the **Name** field, enter the name of the Account Segment.

- b. In the **ID** field, enter an ID for the new Account Segment.
- c. Click << **Add**.

To remove an Account Segment from the selection list:

- a. Select the account segment from the **Account Segment** selection list.
 - b. Click **Remove>>**.
4. Select an **External ID Type** from the dropdown list.
- Possible choices are:
- ☐ Default - select it to configure the Card Generator to produce Recharge Cards.
 - ☐ 4: Calling Card - select it to configure the Card Generator to produce Calling Cards.
 - ☐ 10: Unified Card - select it to configure the Card Generator to produce Unified Cards.
5. Click **Next>>** to move to the next configuration page.



To enter values on Page Four of Configuration window:



NOTE

A Priority user is defined in the fourth page of the Configuration window.

Figure 14 Configuration Window - Page Four

1. In the **User Name** field, enter the user name of the priority user.
2. In the **Real Name** field, enter the full name of the priority user.
3. In the **Comment** field, enter any comments about the user. This is an optional field.
4. In the **Password** field, enter the priority user's password. The password must contain the minimum number of digits previously set in the **Password Length** field on the first page of the Configuration window.
5. In the **Confirm** field, re-enter the password to confirm its correctness.
6. Click **Finish** to complete the configuration.

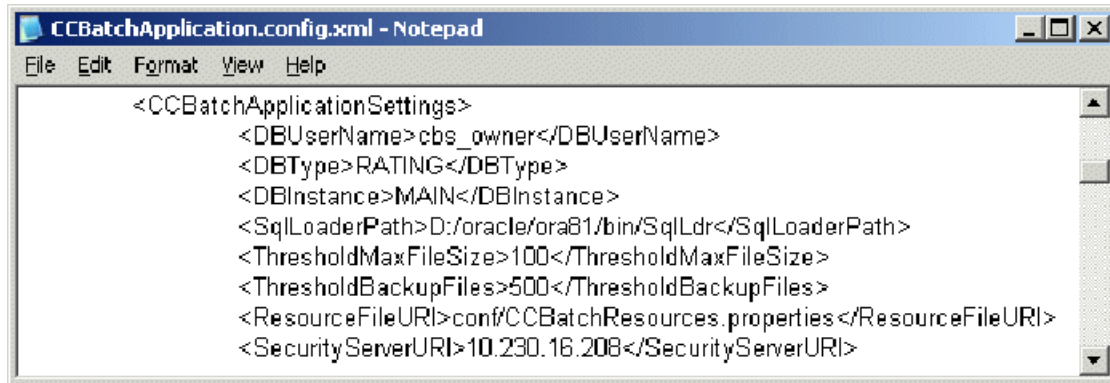
Configuring Customer Care Batch (CCBatch)

To configure Customer Care Batch

1. Configure the TnsNames file for SQL Loader. TnsNames file should have an entry for DB connection with alias MAIN. A valid IP, Port, and SID values should be assigned.
 MAIN=
 (DESCRIPTION =
 (ADDRESS = (PROTOCOL = TCP)(HOST= <IP>)(PORT= <port>))
 (CONNECT_DATA = (SID = <SID>)))

2. Open the configuration file located in <CCBatch Installation Folder>\conf\CCBatchApplication.config.xml, using notepad.
3. Setup the **Database** and **Security Server** in Application Config File.
 - a. Search and navigate to < CCBatchApplicationSettings> tag as shown below

Figure 15 CCBatchApplicationSettings Tags



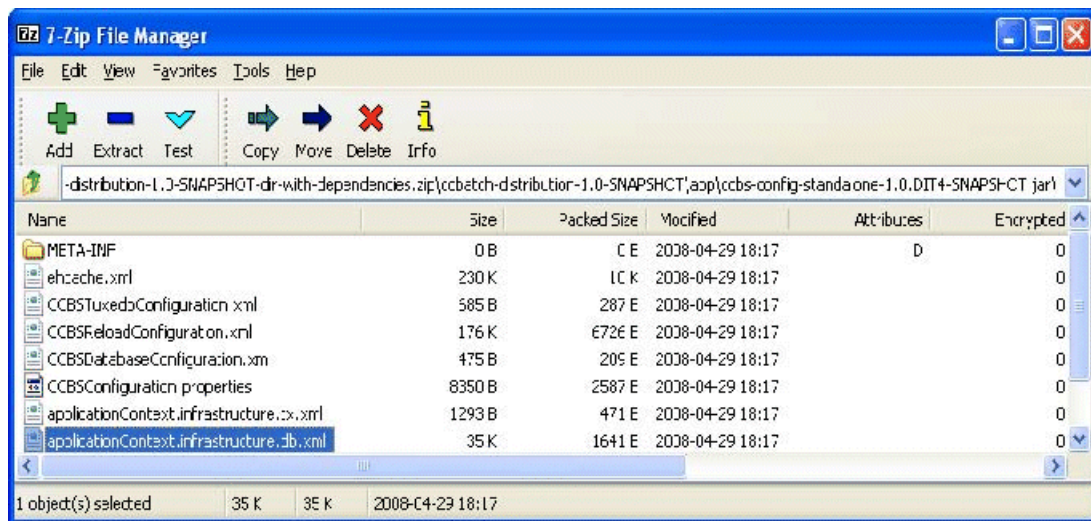
- b. Verify the following information:
 - ☐ The value of DBUserName element is the correct Rating DB User Name.
 - ☐ The value of DBType element is 'RATING'.
 - ☐ The value of DBInstance elements is 'MAIN'.
 - ☐ The IP Address of "SecurityServerURI" element (Security Server) is correct.
 - ☐ The SqlLoaderPath is correct (Oracle 10g SqlLoader path).
 - c. Save the changes to the CCBatchApplication configuration file.
4. Setup the DB IP, UserName, and Password in applicationContext.infrastructure.db.xml file.



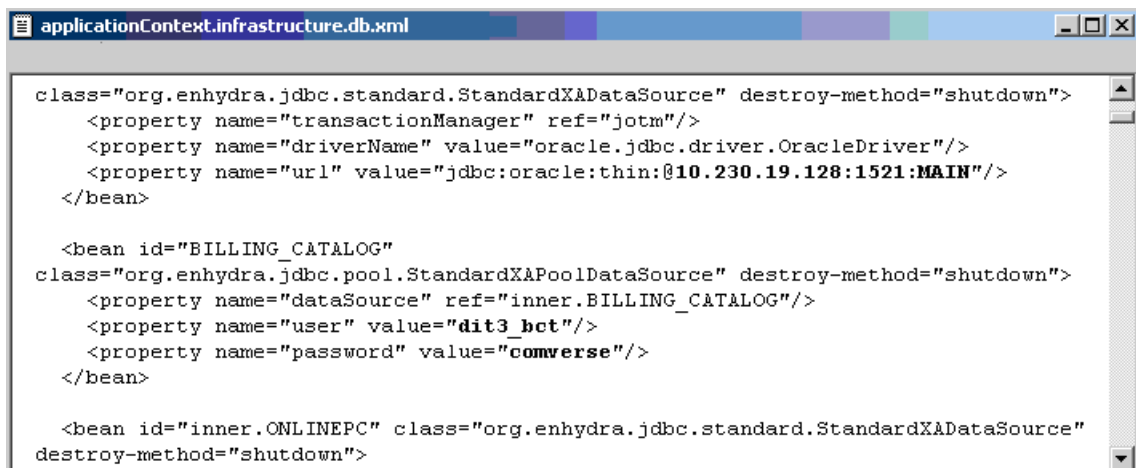
NOTE

To avoid UNIX and DOS file format conflicts, use VI Editor for updating the file.

- a. Open the ccbs-config-standalone-1.0.DIT4-SNAPSHOT.jar file from <CCBatch Installation Folder>\app folder, using 7 Zip File Manager.

Figure 16 7-Zip File Manager

- b. Edit the `applicationContext.infrastructure.db.xml` file in `ccbs-config-standalone-1.0.DIT4-SNAPSHOT.jar` file.

Figure 17 Editing `applicationContext.infrastructure.db.xml` File

- c. For all the DB entries, assign the correct IP Address, SID, UserName, and Password.
- d. To align with SAPI framework change for running with a separate CVS, the file has to be updated with JNDI information for CVS. The JNDI entry for Standalone CVS mentioned below has to be added in the file.

```

<bean id="inner.RATING_VCDB"
class="org.enhydra.jdbc.standard.StandardXADataSource" destroy-
method="shutdown">
  <property name="transactionManager" ref="jotm" />
  <property name="driverName" value="oracle.jdbc.driver.OracleDriver" />
  <property name="url" value="jdbc:oracle:thin:@milsyt09.kenan.com:1521:CBS3004"
/>

```



```

</bean>
<bean id="RATING_Vcdb"
class="org.enhydra.jdbc.pool.StandardXAPoolDataSource" destroy-
method="shutdown">
<property name="dataSource" ref="inner.RATING_Vcdb" />
<property name="user" value="drop31_tc10_dev" />
<property name="password" value="arbor123" />
</bean>

```

And, below key need to be added in the DataSourceLookup to map the datasource

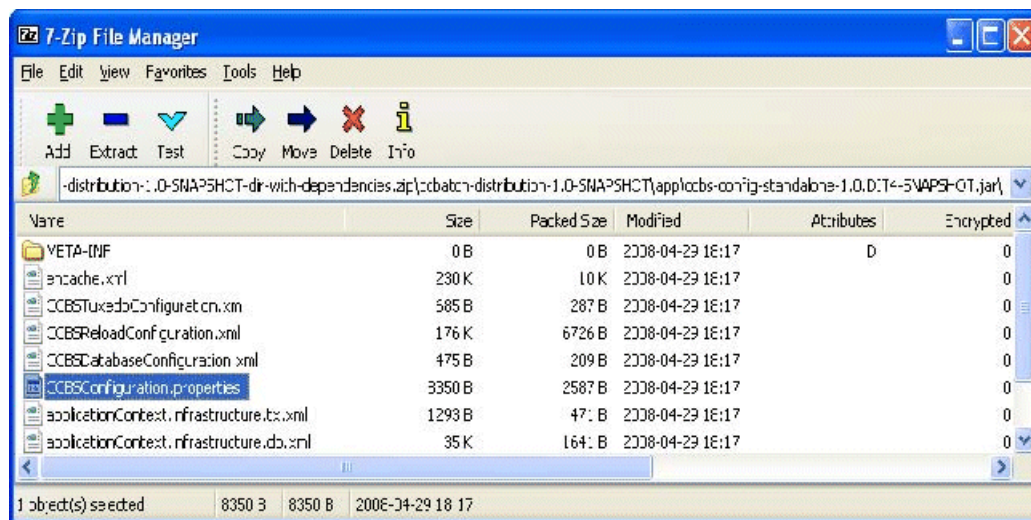
```

<entry key="jdbc/RATING_Vcdb">
<ref bean="RATING_Vcdb" />
</entry>

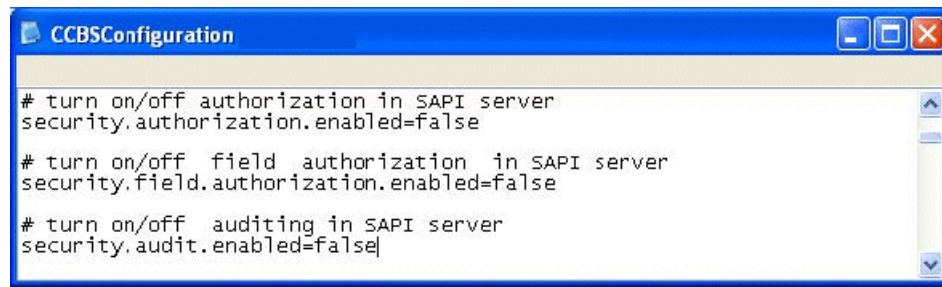
```

- e. Save the changes in applicationContext.infrastructure.db.xml file and close Zip File Manager.
5. Setup the **Security** and **TimeOut** Properties.
- a. Open the ccbs-config-standalone-1.0.DIT4-SNAPSHOT.jar file from <CCBatch Installation Folder>\app folder, using 7 Zip File Manager.

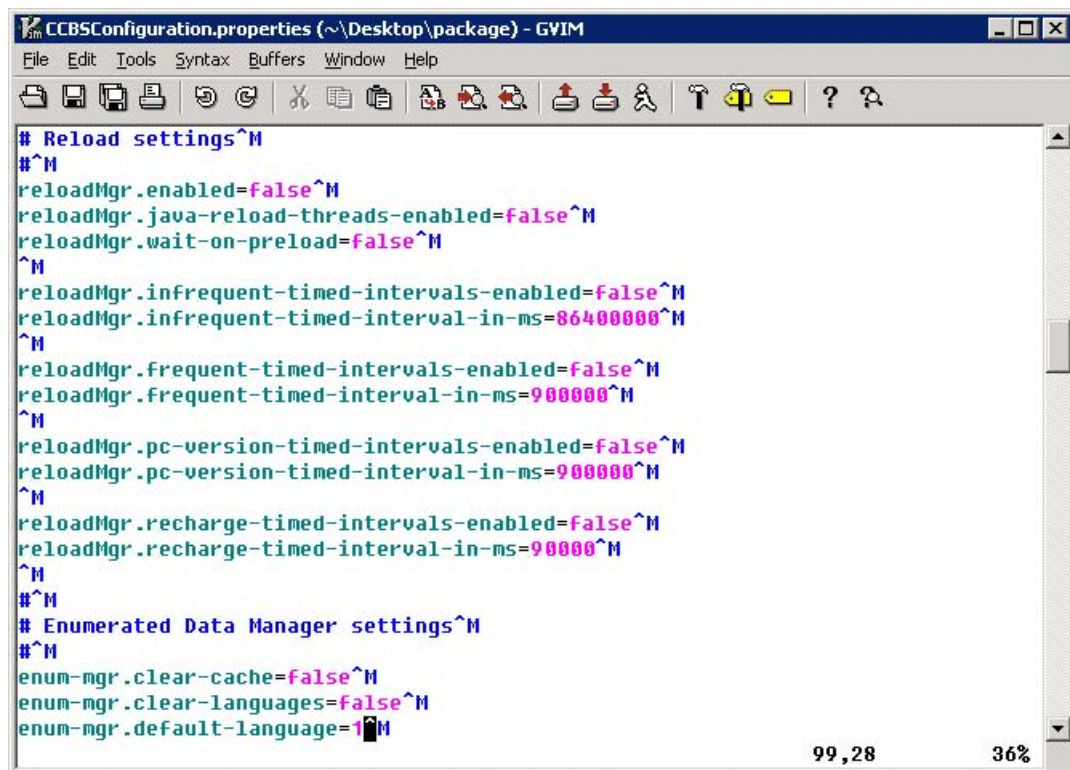
Figure 18 7-Zip File Manager



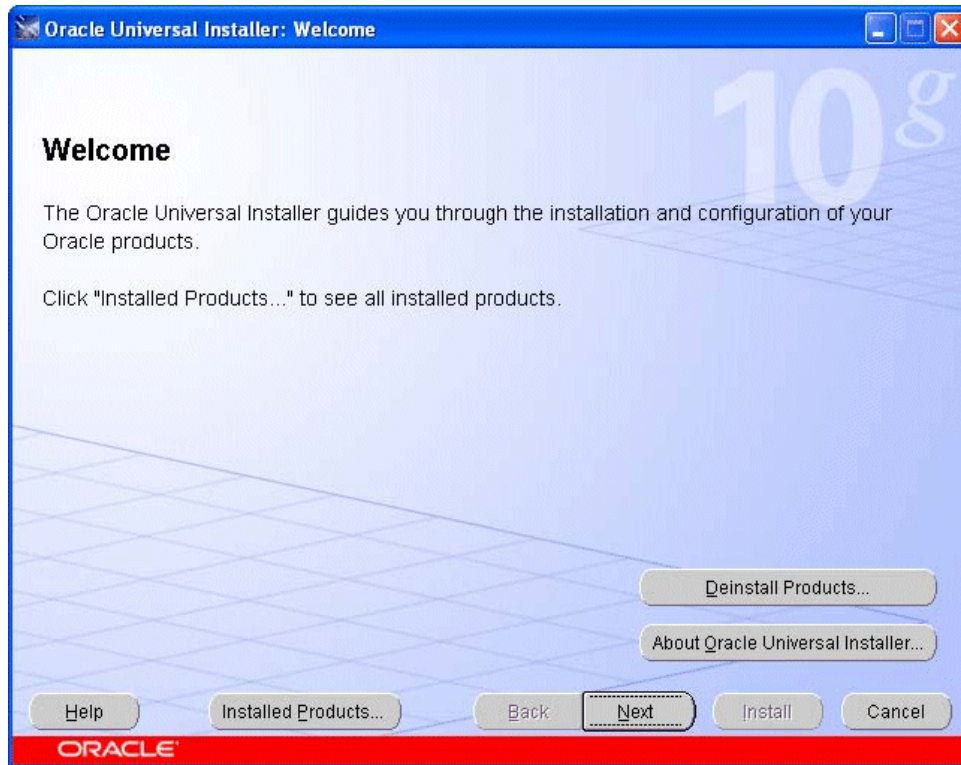
- b. Edit the CCBSCConfiguration.properties file in ccbs-config-common-shared-1.0.DIT4-SNAPSHOT.jar file.

Figure 19 Editing CCBSSConfiguration.properties File

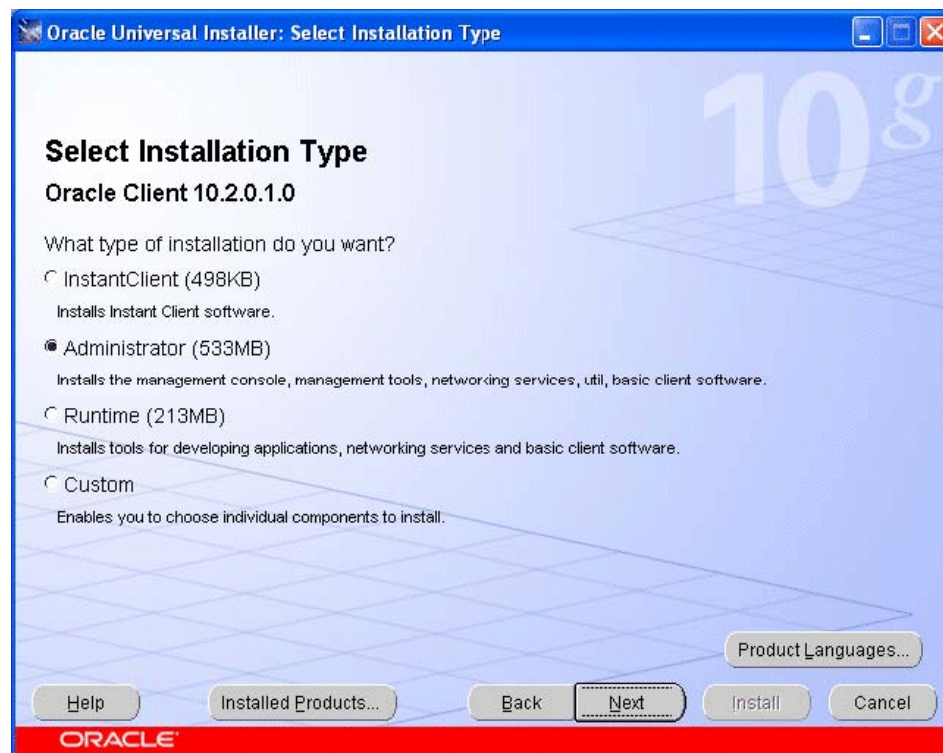
- c. Turn off the authorization, field authorization, and auditing settings.
- d. Update the following parameters:
 - reloadMgr.enabled=false
 - reloadMgr.java-reload-threads-enabled=false
 - reloadMgr.wait-on-preload=false
 - reloadMgr.infrequent-timed-intervals-enabled=false
 - reloadMgr.infrequent-timed-interval-in-ms=86400000
 - reloadMgr.frequent-timed-intervals-enabled=false
 - reloadMgr.frequent-timed-interval-in-ms=900000
 - reloadMgr.pc-version-timed-intervals-enabled=false
 - reloadMgr.pc-version-timed-interval-in-ms=900000
 - transaction.timeout.default=600

Figure 20 Updating CCBSSConfiguration.properties Parameters

- e. Save the changes in CCBSSConfiguration.properties file and close Zip File Manager.
6. Install the SQL Loader.
 - a. Start Oracle 10g Client installation.

Figure 21 Oracle 10g Client Installation Home Window

- b. Select Administrator installation type.

Figure 22 Installation Type Window

- c. Make note of client installation path.

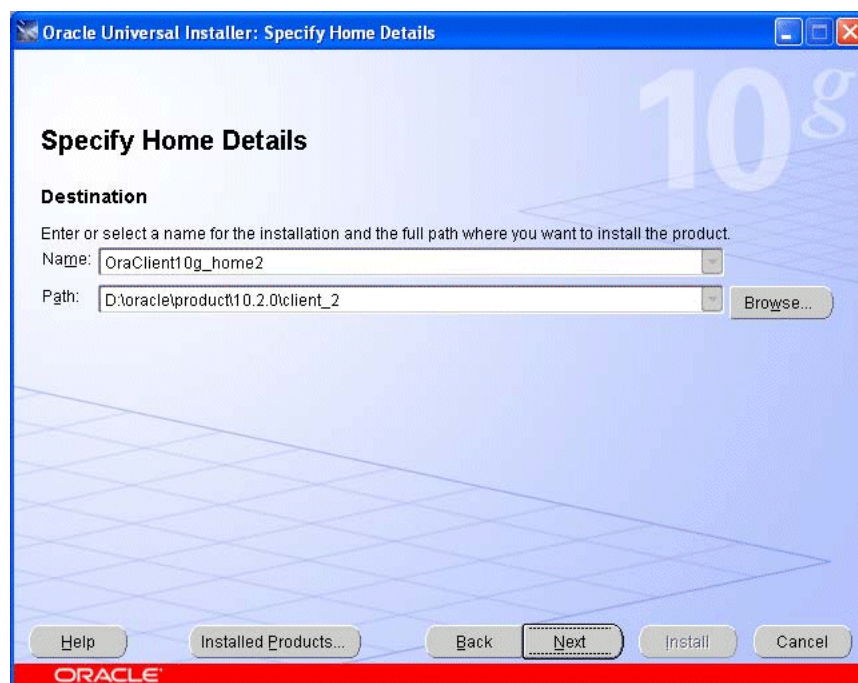
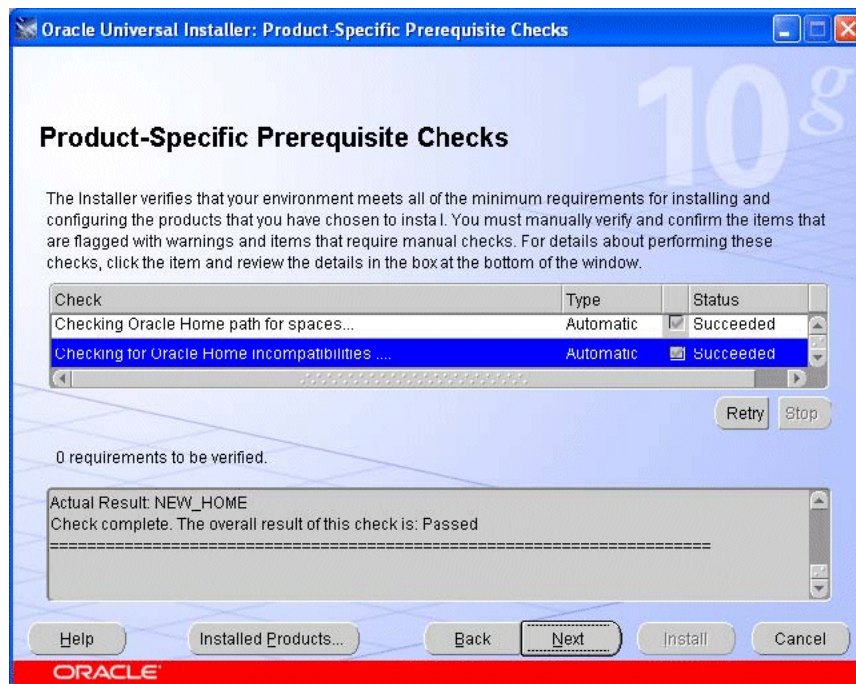
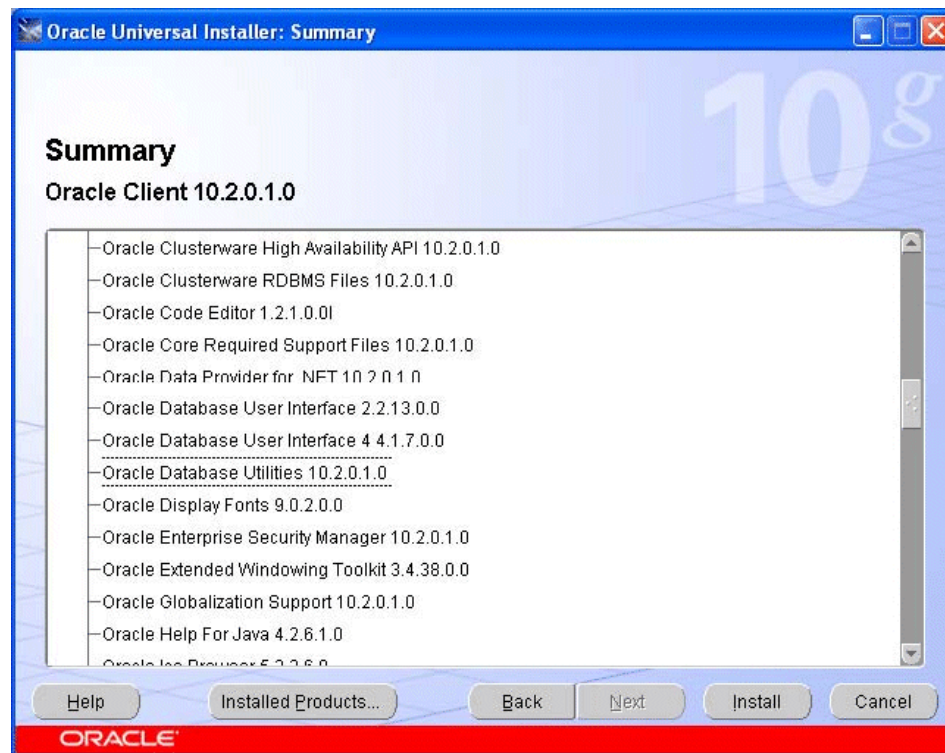
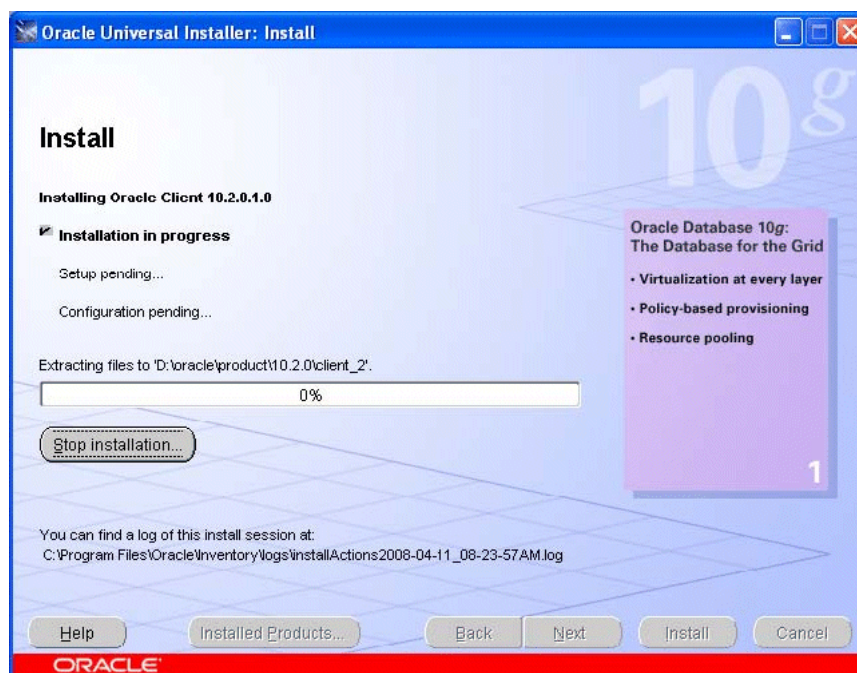
Figure 23 Specify Home Details Window

Figure 24 Product-Specific Prerequisite Checks Window

- d. Make sure to have “Oracle Database Utilities 10.2.x.x.x”.

Figure 25 Summary Window

- e. Proceed with the installation.

Figure 26 Oracle Client 10.2.0.1.0 Installation Window

- f. After the installation is complete, check the presence of sqlldr.exe in <Oracle Home>/bin directory.

Optimal Voucher Recharge Process Schedule

The Voucher Recharge process can be run at peak, off-peak or on-demand based on the source of initiation namely IVR, USSD, Self Care, and CSR.

Reviewing Voucher Recharge Process Activity

CCBatch Log file is placed at C:\CCBatch.

Expected Input for Voucher Recharge Process

The Voucher Recharge operations need a recharge card (voucher) and a subscriber or an account identifier.

The interface specifications are mentioned below:

VoucherRechargeResult voucherRechargeAccount (Integer accountNo, SubscriberIdentifier subscriberId, String secretNumber, Short unitType, Double sessionRecharge, Integer originatingApp, String comments)

VoucherRechargeResult voucherRechargeSubscriber (SubscriberIdentifier subscriberId, String secretNumber, Short unitType, Double sessionRecharge, Integer originatingApp, String comments)

Where,

- **accountNo** - Account Identifier
- **subscriberId** - Subscriber Identifier
- **secretNumber** - Voucher's secret number
- **unitType** - Not used (can be null) currently
- **sessionRecharge** - used to detect and enforce the session recharge limits
- **originatingApp** - ID of the calling application (from the RechargeOriginator enumeration)
- **comments** - free text

For example:

```
VoucherRechargeClient vrc = ApplicationContext.get (VoucherRechargeClient.class);
vrc.voucherRechargeSubscriber (subscriber, "14242847287424", (short) 0, 0d, 1, "a simple
recharge");
```

where, subscriber is a valid subscriber identifier.

Expected Output and Location of Output

The recharge APIs return the VoucherRechargeResult object. It contains the list of the changed balances (DeltaBalance) and the additional return value flags and statuses from the recharge operation.

List of Success/Failure Criteria for Voucher Recharge Process

The operation is successful if the recharge attempt succeeds. The APIs throw exceptions for the encountered errors. If the recharge is successful but some additional operations fail (for example, PO swap), SAPI returns the value as true in VoucherRechargeResult.
rechargeSucceededWithErrors.

Troubleshooting Guidelines

CCBatch Troubleshooting

The Developer Log can be turned to debug mode by Tier-4 person to trouble shoot the application in case of any failures. By default, the Developer Log is set to error mode.



To change the Developer Log Mode

1. Close the CCBatch application if it is running.
2. Navigate to log configuration file in the jar located in <CCBatch Application Installation Folder>\app\ccbatch-app-<version number>.jar'.
3. Edit the Log4j.xml file.
4. Search for the **com.comverse** logger name. Change the element level under **com.comverse** logger to **debug**.

```
<logger name="com.comverse">
```

```
    <level value="debug" />
```

```
    <appender-ref ref="rolling_file_appender" />
```

```
    <appender-ref ref="console_appender" />
```

```
</logger>
```

5. Save and close the Log4j.xml file.

Chapter 3

CCAP Process

3

orded (sender
The desti
notifying
ng The noti
ieve The m
t access To

e

CCAP Process Description

CCAP (Offline URE) is used to process both switch CDR file and outage record file in the offline mode. It also processes re-rating file and error corrected file.

Prerequisites for Running CCAP Process

Before starting CCAP process, the TSP process must start successfully in offline mode on the same SLU where the CCAP runs and the UPA is also started because CCAP needs to connect to both.

Dependencies on Other Processes

CCAP processes files generated by CMCAP, MIUB, URC, or LTP. CCAP does not do anything if there are no files for it to process.

Running CCAP Process

1. Run from UPM.
2. Run from Command Line.

CCAP can be started from the command line and CCAP is expected to run as follows in the production mode:

```
% cd /home/omni/bin
```

```
% ./ure -C -ccapname <process_name> -ratingdb <rating DB id> -instance <instance_id> -log <log file name>
```

Where,

- ❑ **<process_name>** is used to populate PROCESS_SCHED.process_name when scheduling the offline Rater task. The max length for PROCESS_SCHED.process_name is 8, so make sure the process_name does not exceed the max length. The <process_name> must be unique for each running offline URE process.
- ❑ **<rating DB id>** is the rating DB ID (for example, 9) on which the CCAP runs.
- ❑ **<instance_id>** is the instance ID (for example, 1) for the given CCAP process and should be different for each running CCAP process on the same rating DB.
- ❑ **<log file name>** is the log file name (for example, CCAP_logfile) for CCAP's log messages. The log file is stored under the directory specified by the parameter **<LogFilePath>** in ureCfg.xml.

CCAP is usually started from UPA in production.

Configuration of CCAP Process

For running CCAP, the SLU must be configured with the configureOFR utility during the Offline URE package installation.

\$ORACLE_HOME/network/admin/tnsnames.ora must have appropriate entries to access the Rating DB, calling circle DB and the Billing Admin (unscaled or ORP) DB.

The rating DB TNS entry has this format: **sdp<rating DB id>_n<node #>.world**.

For example, **sdp9_n1.world** is for rating DB 9, node 1.

The calling circle DB TNS entry has this format: **ccd<calling circle DB id>_n<node #>.world**.

For example, **ccd1_n1.world** is for calling circle DB 1, node 1.

The XML configuration files under /home/omni/conf are required to be configured as follows:

commonCfg.xml: Update **<DefaultDatabaseInstance>** for rating DB instance, **<DefaultDatabaseType>** rating DB type and **<DefaultOracleUserName>** for the DB user name based on the provisioned data in the Security Server. The configured data is used to get the rating DB password from the Security Server.

Example:

```
<DefaultDatabaseInstance>MAIN</DefaultDatabaseInstance>
<DefaultDatabaseType>RATING</DefaultDatabaseType>
<DefaultOracleUserName>cbs_owner</DefaultOracleUserName>
```

tspCfg.xml: Update **<DEFAULT_SDP_NODE>** for the default rating DB instance to load configuration data.

Example:

```
<DEFAULT_SDP_NODE>sdp9_n1.world</DEFAULT_SDP_NODE>
```

ureCfg.xml: Update **<AdminDbName>** for the billing DB instance, **<BILLING_DB_TYPE>** for the billing DB type and **<BILLING_DB_USER_NAME>** for the DB user name based on the provisioned data in the Security Server. The configured data is used to get the billing DB password from the Security Server. Update **<AdminDbType>** to 4 for Unscaled DB or 12 for ORP DB.

Example:

```
<AdminDbName>blus</AdminDbName>
<AdminDbType>4</AdminDbType>
<BILLING_DB_TYPE>BILLING</BILLING_DB_TYPE>
<BILLING_DB_USER_NAME>cbs_owner</BILLING_DB_USER_NAME>
```

CCAP gets the DB password from the security server. Therefore the security server needs to be specified in /etc/hosts.

Example:

```
10.230.20.88  secserv
```

An entry for the SLU where the CCAP runs must be present in the HOST_CONTACTS in the admin DB and the corresponding directories specified for this entry must be created with the user running CCAP on the SLU.

Example:

Create /data/ccap/usage/ready, /home/sncpuser/ccap/usage/output, /data/ccap/usage/done, /data/ccap/usage/error, and /data/ccap/usage/work directories and insert a row into HOST_CONTACTS (assuming IP address for the SLU is 10.230.18.216)

```
insert into HOST_CONTACTS(HOST_CONTACT_ID, HOST, HOST_MODULE, HOST_
LOGIN, HOST_PASSWORD,
  HOST_READY, HOST_READY, HOST_DONE, HOST_WORK, HOST_ERROR, ARBOR_
DETAIL, HOST_PROCESS_DELETE, HOST_FTP_DELETE, CHANGE_WHO, CHANGE_DT,
HOST_PROTOCOL)
values (99, '10.230.18.216', NULL, 'sncpuser', 'sncpuser', '/data/ccap/usage/ready',
'/data/ccap/usage/output', '/data/ccap/usage/done', '/data/ccap/usage/work',
'/data/ccap/usage/error', NULL, 'N', 'N', 'Test', SYSDATE, 'sftp');
```

A task must be scheduled (inserting a row in PROCESS_SCHED on admin DB) for the CCAP instance using its process name every time before starting CCAP process. The CCAP process name is specified with argument **-ccapname <process_name>** on the URE command line.

Example:

-ccapname ccap01 and a non-recurring CCAP task can be scheduled as follows:

insert into PROCESS_SCHED (process_name, task_name, task_cycle, task_mode, sched_start, task_intrvl, task_status, task_priority, slide_time, db_name, sql_query, debug_level, usg_version) values

```
('ccap01','ccap01','N',1,SYSDATE - 1, 60, 0, 1, 0, NULL, 1, 0);
```

Environmental Variables

The required environment variables are created while running configureOFR. You may edit /home/sncpuser/profile/omnisetup.sh to set env vars as appropriate. LD_PRELOAD must be set to /home/omni/library/libcrypto.so (or whatever is the full path for libcrypto.so) if **sftp** protocol is used for file transfer.

System Parameters

Some of the offline URE log messages are controlled with the URE/LOGLEVEL system parameter. To get full traces, CHAR_VALUE for the parameter should be set to **HIGH**. The default is **MEDIUM**.

Set INT_VALUE for the RTNG/OFFLINE_CDR parameter to 1 if RTB style CDR file needs to be generated for offline usage files. The default value is 0.

Optimal CCAP Process Schedule

CCAP can be scheduled to run at any time depending on the operator's schedule. It usually runs for offline usage during off-peak hours but may need to process the OR files as soon as possible.

Reviewing CCAP Process Activity

Log file name has to be provided on the CCAP command line as mentioned in Running CCAP Process section and the log file (timestamp is appended) is stored under the directory specified by the **<LogFilePath>** parameter in the ureCfg.xml (the default log directory is /home/omni/Logs).

Expected Input for CCAP Process

Usually CMCAP sends the CCAP files for processing but MIUB, URC, and LTP can also generate files for CCAP to process.

Expected Output and Location of Output

CCAP successfully inserts the rated records inUSAGE_RECORD_MAIN (Converged mode) or history tables (Real Time mode) on the rating DB.

CCAP generates output data files (error file, redirected file, re-rating files, and so on.) under the directory defined in HOST_CONTACTS.host_output for the SLU's entry. CCAP also generates CDR files under /data/<slu name> if it is configured to do so.

List of Success/Failure Criteria for CCAP Process

A usage file is considered to be processed successfully by CCAP if the records in the file are either inserted into the rating DB or written into the CCAP output file. Otherwise the file is not considered for processing. The processing status is available in the CCAP log file for each processed file.

Troubleshooting Guidelines

Check the log files for messages and make correction if required.

List of common messages and action to correct

- No task scheduled: Schedule a task as mentioned in Configuration of CCAP Process section.
- Failed to get password from security server for user ...: Ensure that the ureCfg.xml has the correct DB information and the credentials are provisioned in the Security Server.
- Missing directories: ...: Check if the directory path exists and is same as the defined path in the HOST_CONTACTS for the SLU.
- No read/write access: ...: Grant read/write permissions to the directories.
- DB error or another CCAP with the same process_name is running: An entry already exists in PROCESS_STATUS. Ensure that no other CCAP process with the same process_name is running. If the entry is left by the previous crashed process, remove it.

Maintenance Guidelines

File System Maintenance

The new tool UFP may be used to purge files that have been successfully processed by CCAP.

Chapter 4

C-MCAP Process

4

orded (sender
The desti
notifying
ng The noti
ieve The m
cT access To

e

v

C-MCAP Description

C-MCAP has the following four primary roles.

1. Routing/Segmenting data to the appropriate DB for processing
2. Detection of duplicate usage records from the same or different input files
3. Setting up various audit trail records in the appropriate databases
4. Parsing data from the raw incoming format into a common record format and appending some relevant data to those records. This is a new role of C-MCAP relative to MCAP.

Prerequisites for Running C-MCAP Process

COM must be run before running C-MCAP. COM transfers usage files from external sources and places them into the appropriate location for C-MCAP (EXT_CONTACTS.arbor_ready). COM also inserts a row into FILES_TO_PROCESS, and FILE_STATUS and therefore C-MCAP can query these tables to get the list of files required to be processed.

Dependencies on Other Processes

If COM is not running before C-MCAP, C-MCAP simply terminates without doing any work.

Running C-MCAP Process

Run from UPM

1. To start C-MCAP process (in Converge Env and in batch mode):
`execute_process -g application -pr cmcap01 -d start -t I -bp CMCAP -r "task_name=cmcap01, server_id=4, task_mode=0"`
2. To start C-MCAP process (in Converge Env and in continuous mode):
`execute_process -g application -pr cmcap01 -d start -t I -bp CMCAP -r "task_name=cmcap01, server_id=4, task_mode=0, task_cycle=R, task_intrvl=60"`
3. To start C-MCAP process (in Real-time Env and in batch mode):
`execute_process -g application -pr cmcap01 -d start -t I -bp CMCAP -r "task_name=cmcap01, server_id=12, task_mode=0"`
4. To start C-MCAP process (in Real-time Env and in continuous mode):
`execute_process -g application -pr cmcap01 -d start -t I -bp CMCAP -r "task_name=cmcap01, server_id=12, task_mode=0, task_cycle=R, task_intrvl=60"`

Where,

 - ❑ **task_cycle** = R: Run in recurring mode.
 - ❑ **task_intrvl** = 60: Sleep Interval (One minute).
 - ❑ **server_id** = 4: Server ID of BLUS in Converge env.
 - ❑ **server_id** = 12: Server ID of ORP in Real-time mode.
 - ❑ **task_mode** = 0: Production mode. COM always runs on this mode.
5. To stop C-MCAP process:
`execute_process -g application -pr cmcap01_4 -d shutdown` (for CMCAP run on Converge env)
`execute_process -g application -pr cmcap01_12 -d shutdown` (for CMCAP run on Realtime env)

Run from Command Line

CMCAP <process_name> <Unscaled server ID> (i.e. CMCAP com01 4)

Example:

CMCAP cmcap01 4 (run CMCAP on Converge env)

CMCAP cmcap01 12 (run CMCAP on Real-time env)

Where,

- **Process_name:** Name of the Process to be run. For example, cmcap01, cmcap02, cmcap03, and so on.
- **Unscaled server ID:** Server ID of BLUS (on Converge) or ORP (on Real-time) DB.

Configuration of C-MCAP Process

No configuration file. The configuration is done in the database.

Environmental Variables

- **ARBORDIR:** Top level directory where the apps are installed.
- **ARBORLOG:** The path where the log file must be placed (/staging/billing/log).
- **ARBORCTRLRPT:** The path where the control report file must be placed (/staging/billing/reports).
- **ARBOR_CATALOG_DATABASE:** Catalog database server (CTLG for Converge, MAIN for Real-time mode).
- **ARBOR_CATALOG_QUERY:** Catalog database instance (CTLG for Converge, MAIN for Real-time mode).
- **ARBORDATA:** Top level directory where the usage files are located (/staging/billing/data).
- **ARBORBIN:** All binaries are located.
- **ARBORDBU:** User schema to log in database (that is, cbs_owner).
- **OAM_ENV_CONN_MA:** OAM_ENV_CONN_MA = TRUE to connect to the Security server to get the database password and OAM_ENV_CONN_MA = FALSE for not using Security server to get the database password.

System Parameters

The system parameters are:

- **MCAP/LOGLEVEL**
 - LOW // least log information
 - MEDIUM // medium log information
 - HIGH // most log information
- **MCAP/TRA_SWITCH**
 - 0 // turn off SQL log
 - 1 // turn on SQL log
- **MCAP/TABLE_SPACE**
 - // Populate one of the available table space in the admin DB (BLUS or ORP)
- **MCAP/FIELD_DELIMITER** // when process outage record

- Default = "|", a pipe
- MCAP/CDR_DUPLICATE_CHECK
 - 0 // Do not perform the dup check
 - 1 // Do record duplicate checking
- Admin Tables

EXT_CONTACTS

insert into EXT_CONTACTS

(ext_contact_id, is_send, source_id, ext_client_id, description_code, access_method, respond_to_id, ext_ebcdic, prefix_match, postfix_match, unique_ext_name, arbor_ready, arbor_done, arbor_error, arbor_work, max_time_gap, retry_delay, retry_limit, active_dt, inactive_dt, change_who, change_dt, control_file_expected, is_rfr_required, rfr_path, rfr_filename, rfr_delay, file_type, arbor_detail, use_raw_usage, used_by_server_type)

values

(122, 0, 1, 111, 1, 1, NULL, 0, NULL, NULL, 1, '/staging/billing/data/usage/ready', '/staging/billing/data/usage/done', '/staging/billing/data/usage/error', '/staging/billing/data/usage/work', NULL, NULL, NULL, '01-01-1990', NULL, 'cbs_owner', '\$DATE', 0, NULL, NULL, NULL, NULL, 0, '/staging/billing/data/usage/detail', 0, 4);



- To do file duplicate checking -> set **unique_ext_name = 1**
- To run on ftp/sftp mode -> set **access_method = 2**
- To run on ORP DB -> set **used_by_server_type = 12**

HOST_CONTACTS

insert into HOST_CONTACTS (host_contact_id, host, host_module, host_login, host_password, host_ready, host_work, host_done, host_error, arbor_detail, host_process_delete, host_ftp_delete, change_who, change_dt, failover_host, host_protocol, host_output)

values

(122, 'Local machine IP address', 'CMCAP', 'oracle8', 'oracle8', '/staging/billing/data/usage/ready', '/staging/billing/data/usage/work', '/staging/billing/data/usage/done', '/staging/billing/data/usage/error', '/staging/billing/data/usage/detail', 'Y', 'Y', 'oracle8', SYSDATE, 'failover host IP address', 'ftp', '/staging/billing/data/usage/ready');

- CDR_DUP_CKECK_FIELDS // used in Converge mode
- Configure this table when the system parameter MCAP/CDR_DUPLICATE_CHECK turns on.

- **ORP_DUP_CKECK_FIELDS** // used in Converge/Real time mode
Configure this table when the system parameter **MCAP/CDR_DUPLICATE_CHECK** turns on.

Optimal C-MCAP Process Schedule

This process runs on demand.

Reviewing C-MCAP Activity

The log file is located under \$ARBORLOG directory. The Control report is located under \$ARBORCTRLRPT directory.

Example: Log filename convention

C-MCAP control report file: MCAP08-090320-171140-04-860162.rpt

C-MCAP log file: mcap08-090320-171109-04-860162.BATCH.000

Expected Input for C-MCAP Process

Offline usage files and OR files brought into the billing system by COM are the inputs for C-MCAP process. Before running CMCAP, it has to be scheduled to run by inserting the following SQL to PROCESS_SCHED table into the unscaled database:

```
insert into PROCESS_SCHED
```

```
(process_name, task_name, task_cycle, task_mode, sched_start, task_intrvl, task_status,  
task_priority, slide_time, db_name, sql_query, debug_level, plat_id, usg_crt_hour, usg_plat_id,  
usg_version)
```

```
VALUES ('cmcap01', 'cmcap01', 'N', 0, SYSDATE, 60, 0, 2, 55, 'BILL',  
'EXT_CONTACTS.ext_contact_id=122', 1, NULL, 0, NULL, 0);
```

Where,

- **Process_name:** Name of the process to run (cmcap01, cmcap02, cmcap03 and so on.).
- **Task_name:** Usually it is the same as the process_name.
- **Task_cycle:** Recurring mode (R) or Non-recurring mode (N).
- **Task_mode:** Equal to 0 for production mode.
- **Sched_start:** Schedule time to start the process.
- **Task_intrvl:** Applicable only in recurring mode. The sleeping time between each run (Default Value = 60 seconds).

Command to Run

```
CMCAP cmcap01 <unscaled server ID>
```

Unscaled server ID = 4 if run on BLUS (CV mode)

Unscaled server id = 12 if run on ORP (CV/RT mode)

Expected Output and Location of Output

C-MCAP places the output usage files into the location specified in **HOST_CONTACTS.host_output** for **host_module** equal to C-MCAP, for output usage files. C-MCAP inserts a row into the

USAGE_FILE_STATUS and USAGE_FILES_TO_PROCESS tables on the unscaled database and therefore the downstream process (CCAP and LTP) knows from where to take the usage files.

List of Success/Failure Criteria for C-MCAP Process

After complete processing, C-MCAP updates its status on the PROCESS_STATUS table on the unscaled database. Normally, PROCESS_STATUS.process_status = LOGOUT indicates that the process has run successfully.



See the PROCESS_STATUS table in the Converse ONE Database Reference for information on process_status values.

After C-MCAP finishes processing, first take a look at the control report file first. It mentions the details as mentioned in the example below.

Example:

```
ARBOR          MCAP CONTROL REPORT          20 Mar 2009
AUTHOR: cbs_owner          17:13:30
PROGRAM: CMCAP          PAGE: 1
```

Process: mcap08

Process start time: 03/20/09 17:13:30

File ID Srv	Seq	Input	Split	Records	Good	Soft	Dup	Total
ID File name	Num	Records	Records	Added	Records	Errors	Errors	
Records								
-----	-----	-----	-----	-----	-----	-----	-----	-----
2309 4 128.pqa_test.dat		1	0	0	0	1	0	1

If something is wrong in the control report, then take a look at the log file to see what went wrong.

```
20090324 12:36:02.032 (D 0000000) [0][0][001] EMIT_DEBUG
Start gfr_DataFileIOWrite::OpenFile -
/users/denver/arbor/data/data/usage/ready/128.pqa_
test.dat.04.2876.04.04.0.111.0

20090324 12:36:02.034 (D 0000000) [0][0][001] EMIT_DEBUG
End gfr_DataFileIOWrite::OpenFile -
/users/denver/arbor/data/data/usage/ready/128.pqa_
test.dat.04.2876.04.04.0.111.0

=====
```

```

20090324 12:36:02.043 (E 2003420) [0][0][001] MCAP_STAT01
Good Records: 0

20090324 12:36:02.043 (E 2003426) [0][0][001] MCAP_STAT04
Duplicated Records: 0

20090324 12:36:02.043 (E 2003422) [0][0][001] MCAP_STAT02
Soft Error Records: 1

20090324 12:36:02.043 (E 2003424) [0][0][001] MCAP_STAT03
Total Records:1

=====

20090324 12:36:04.721 (I 4003328) [0][0][001] MCAP_FILE_
ACCEPTED File ACCEPTED; Success processing 128.pqa_test.dat
(2876)

```

Troubleshooting Guidelines

Error

Cannot log into database

Solution

1. If the environment OAM_ENV_CONN_MA = TRUE, then check the Security server connectivity.
2. If the environment OAM_ENV_CONN_MA = FALSE, then check if the encrypt password file exists or not.
3. If both 1) and 2) are fine, then check the environments below to see if these are correctly set:
 - ❑ ARBOR_CATALOG_DATABASE
 - ❑ ARBOR_CATALOG_QUERY
 - ❑ ARBORDBU
4. The ARBOR_CATALOG_DATABASE and ARBOR_CATALOG_QUERY environment variables must be set as follows:

For CV mode:

ARBOR_CATALOG_DATABASE = CTLG

ARBOR_CATALOG_QUERY = CTLG

For RT mode:

ARBOR_CATALOG_DATABASE = MAIN

ARBOR_CATALOG_QUERY = MAIN

Error

Could not find type_id_usg: 40012

Solution

The configuration for the corresponding initial_aut_id is missing in the AUT_INITIAL_REF table or the version of the initial_aut_id is out of the range of trans date of the usage record.

Error

Unable to start up -> unable to initialize

Solution

The static data is missing or corrupted on the unscaled database (BLUS or ORP). Turn on the system parameter MCAP/LOGLEVEL = HIGH and check the log file for more information.

Error

No tasks for process mcapxx

Solution

This error is displayed when the user forgets to schedule a task for the application to run. Insert a row into PROCESS_SCHED table before running C-MCAP.

Error

MCAP_BAD_TOTAL (CMCAP cmcap_exec.cpp 2820) Expected 16 records, but 1 in file <file_name> (file_id)

Solution

The number of records in the file does not match with the number specified in the header/trailer. Check the input file and correct the total number of records in the header/trailer .

ErrorError

Cannot log into database

Solution

1. If the environment OAM_ENV_CONN_MA = TRUE, then check the Security server connectivity.
2. If the environment OAM_ENV_CONN_MA = FALSE, then check if the encrypt password file exists or not.
3. If both 1) and 2) are fine, then check the environments below to see if these are correctly set:
 - ❑ ARBOR_CATALOG_DATABASE
 - ❑ ARBOR_CATALOG_QUERY
 - ❑ ARBORDBU

4. The ARBOR_CATALOG_DATABASE and ARBOR_CATALOG_QUERY environment variables must be set as follows:

For CV mode:

ARBOR_CATALOG_DATABASE = CTLG

ARBOR_CATALOG_QUERY = CTLG

For RT mode:

ARBOR_CATALOG_DATABASE = MAIN

ARBOR_CATALOG_QUERY = MAIN

Error

Cannot find type_id_usg: 40012

Solution

The configuration for the corresponding initial_aut_id is missing in the AUT_INITIAL_REF table or the version of the initial_aut_id is out of the range of trans date of the usage record.

Error

Unable to start up -> unable to initialize

Solution

The static data is missing or corrupted on the unscaled database (BLUS or ORP). Turn on the system parameter MCAP/LOGLEVEL = HIGH and check the log file for more information.

Error

No tasks for process mcapxx

Solution

This error is displayed when the user forgets to schedule a task for the application to run. Insert a row into PROCESS_SCHED table before running C-MCAP.

Error

MCAP_BAD_TOTAL (CMCAP cmcap_exec.cpp 2820) Expected 16 records, but 1 in file <file_name> (file_id)

Solution

The number of records in the file does not match with number specified in the header/trailer. Check the input file and correct the total number of records in the header/trailer .

Error

(NULL libarbor_mod_src/arbor_mod_proc.c 991) Proc mcap08, task : error updating PROCESS_STATUS (CMCAP CMCAP.cpp 114) Another CMCAP process with same name (mcap08) is running... (CMCAP CMCAP.cpp 116) OR previous run of CMCAP with this process_name crashed. (CMCAP CMCAP.cpp 117) Please make sure you don't run multiple CMCAP with the same process_name. (CMCAP CMCAP.cpp 118) And clean up PROCESS_STATUS before continue.

Solution

Ensure that multiple C-MCAPs are not running with the same process_name. Clean up PROCESS_STATUS before continuing and then run the following statement.

```
delete PROCESS_STATUS where process_name = 'mcap08';
commit;
```

Error

mcap08 db_err_handler(full text): ORA-12514: TNS:listener does not currently know of service requested in connect descriptor

Solution

Ensure that the tnsnames.ora file is in sync and has all entries specified in SERVER_DEFINITION.

Error

Process mcap08, server 4: CMCAP ERROR logging into Arbor server_id 4

Solution

Usually BLUS database (server_id 4) or ORP database (server_id 12) are down. Ensure that the DB is running.

Error

(NULL libarbor_mod_src/arbor_mod_login.c 1032) ORA-12154: TNS:could not resolve the connect identifier specified

(CMCAP CMCAP.cpp 163) Can't log into cust DB

Solution

On startup, C-MCAP tries to connect to all Cust DB defined in SERVER_DEFINITION. This error occurs when the tnsnames.ora file does not have correct entry for the Cust DB. Ensure that the tnsnames.ora file is in sync and has all entries specified in SERVER_DEFINITION.

Error

mcap08 db_err_handler(full text): ORA-12154: TNS:could not resolve the connect identifier specified

Process mcap08, server 12: CMCAP ERROR: Unable to login to Catalog DB

Solution

1. Ensure that the tnsnames.orafile is in sync and has all entries specified in SERVER_DEFINITION.
2. Ensure that the ARBOR_CATALOG_DATABASE and ARBOR_CATALOG_QUERY environment variables are set to the correct catalog database (CTLG for CV mode or MAIN for CV/RT mode)

Error

EMIT_DEBUG Unknown record type! Unable to continue.

Solution

The input file is corrupted or is in the wrong format due to which the C-MCAP fails to parse the file. Check the content of the input file and correct it.

Maintenance Guidelines

File System Maintenance

The file system **/staging** and the directory defined in EXT_CONTACTS.arbor_done (usually it is **/staging/billing/data/usage/done**) has to be checked. The usage files in this directory are those files that are successfully processed. Once /staging is almost full, you can safely remove the files under this directory.

Database Maintenance

A utility module name UFA is used to archive/purge the following tables:

- FILE_STATUS
- USAGE_FILE_STATUS
- USAGE_FILE_STATUS_HISTORY
- USAGE_FILE_COUNTS
- USAGE_FILE_COUNTS_HISTORY
- USAGE_LOCAL_COUNTS

Archiving Maintenance

Run UFA once per two weeks or once per month.



See chapter [UFA Details](#) for more information.

Chapter 5

COM Process

5

orded (sender
The desti
notifying
ng The noti
ieve The m
t access To



COM Process Description

COM resides on the unscaled server and transfers the files of usages and external systems such as network usage recorders, credit card clearinghouses, and so on. For example, COM is responsible to place the usage files to a location from which CMCAP can route them to the correct Rating DB for CCAP.

Prerequisites for Running COM Process

The prerequisites to run COM are:

- For retrieving files:
 - In local mode - it is expected that the usage files from the mediation must be placed into the staging area as configured in EXT_LOCAL_DIR_ACCESS.remote_ready.
 - In ftp mode - it is expected that the usage files from the external system must be placed into the area as configured in EXT_FTP_ACCESS.remote_ready.
- For sending files, it is expected that the usage files from the local system must be placed into the staging area that is mentioned in FILE_STATUS table.

Dependencies on Other Processes

No dependencies on other Processes.

Running COM Process

Run from UPM

1. To start COM process (in Converge Env and in batch mode):
`execute_process -g application -pr com01 -d start -t I -bp COM -r "task_name=com01, server_id=4,task_mode=0"`
2. To start COM process (in Converge Env and in continuous mode):
`execute_process -g application -pr com01 -d start -t I -bp COM -r "task_name=com01, server_id=4,task_mode=0,task_cycle=R,task_intrvl=60"`
3. To start COM process (in Real-time Env and in batch mode):
`execute_process -g application -pr com01 -d start -t I -bp COM -r "task_name=com01, server_id=12,task_mode=0"`
4. To start COM process (in Real-time Env and in continuous mode):
`execute_process -g application -pr com01 -d start -t I -bp COM -r "task_name=com01, server_id=12,task_mode=0,task_cycle=R,task_intrvl=60"`

Where,

- **task_cycle** = R: Run in recurring mode.
 - **task_intrvl** = 60: Sleep Interval (One minute).
 - **server_id** = 4: Server ID of BLUS in Converge env.
 - **server_id** = 12: Server ID of ORP in Real-time mode.
 - **task_mode** = 0: Production mode. COM always runs on this mode.
5. To stop COM process:
`execute_process -g application -pr com01_4 -d shutdown` (for COM run on Converge env)
`execute_process -g application -pr com01_12 -d shutdown` (for COM run on Realtime env)

Run from Command Line

COM <process_name> <Unscaled server ID> (that is, COM com01 4)

Example:

COM com01 4 (run COM on Converge env)

COM com01 12 (run COM on Real-time env)

Where,

- **Process_name**: Name of the Process to be run. For example, com01, com02, com03, and so on.
- **Unscaled server ID**: Server ID of BLUS (on Converge) or ORP (on Real-time) DB.

Configuration of COM Process

No configuration file. The configuration is done in the database.

Environmental Variables

- **ARBORDIR**: Top level directory where the apps are installed.
- **ARBORLOG**: The path where the log file must be placed (/staging/billing/log).
- **ARBORCTRLRPT**: The path where the control report file must be placed (/staging/billing/reports).
- **ARBOR_CATALOG_DATABASE**: Catalog database server (CTLG for Converge, MAIN for Real-time mode).
- **ARBOR_CATALOG_QUERY**: Catalog database instance (CTLG for Converge, MAIN for Real-time mode).
- **ARBORDATA**: Top level directory where the usage files are located (/staging/billing/data).
- **ARBORBIN**: All binaries are located.
- **ARBORDBU**: User schema to log in database (that is, cbs_owner).
- **OAM_ENV_CONN_MA**: OAM_ENV_CONN_MA = TRUE to connect to the Security server to get the database password and OAM_ENV_CONN_MA = FALSE for not using Security server to get the database password.

System Parameters

The system parameters are:

- **COM/LOGLEVEL**
 - LOW // least log info
 - MEDIUM // medium log info
 - HIGH // most log info
- **COM/TRA_SWITCH**
 - 0 // turn off SQL log
 - 1 // turn on SQL log
- **COM/SKIP_REMOTE_VERIFY**
 - 0 // verify the existing remote file/directory
 - 1 // skip the verification of the existing remote file/directory

- COM/FTP_USE_FTPTRANSFER
 - 0 // use scripting to do ftp
 - 1 // use ftp_transfer library to do ftp/sftp
- COM/DELETE_REMOTE_FILE
 - 0 // Do not delete remote file after transfer - use in ftp/sftp mode
 - 1 // Delete remote file after transfer - use in ftp/sftp mode
- Admin tables:
 EXT_CONTACTS
 insert into EXT_CONTACTS
 (ext_contact_id, is_send, source_id, ext_client_id, description_code, access_method,
 respond_to_id,
 ext_ebcdic, prefix_match, postfix_match, unique_ext_name, arbor_ready, arbor_done,
 arbor_error,
 arbor_work, max_time_gap, retry_delay, retry_limit, active_dt, inactive_dt, change_who,
 change_dt,
 control_file_expected, is_rfr_required, rfr_path, rfr_filename, rfr_delay, file_type, arbor_
 detail, use_raw_usage,
 used_by_server_type)
 values
 (122, 0, 1, 111, 1, 1, NULL, 0, NULL, NULL, 1,
 '/staging/billing/data/usage/ready',
 '/staging/billing/data/usage/done',
 '/staging/billing/data/usage/error',
 '/staging/billing/data/usage/work',
 NULL, NULL, NULL, '01-01-1990', NULL, 'cbs_owner',
 '\$DATE', 0, NULL, NULL, NULL, NULL, 0,
 '/staging/billing/data/usage/detail', 0, 4);



- To do file duplicate checking -> set **unique_ext_name = 1**
- To run on ftp/sftp mode -> set **access_method = 2**
- To run on ORP DB -> set **used_by_server_type = 12**

```
EXT_LOCAL_DIR_ACCESS
insert into EXT_LOCAL_DIR_ACCESS
(ext_contact_id, remote_ready, remote_work, remote_done)
values
(122, '/staging/billing/data/usage/remote/ready',
'/staging/billing/data/usage/remote/work',
'/staging/billing/data/usage/remote/done');
```

```
EXT_FTP_ACCESS // only required if it is run in ftp/sftp mode
Insert into EXT_FTP_ACCESS (ext_contact_id, remote_host, remote_login, remote_
password, remote_ready, remote_done, remote_work)
```

Values

```
(122,'Remote IP address','oruser','oruser','/or_mount_point/OR/ready','/or_mount_point/OR/done','/or_mount_point/OR/work');
```

Optimal COM Process Schedule

This process runs on demand.

Reviewing COM Process Activity

The log file is located under \$ARBORLOG directory. The Control report located under \$ARBORCTRLRPT directory.

Expected Input for COM Process

Before running COM, it has to be scheduled to run by inserting the following SQL to PROCESS_SCHED table into the unscaled database:

```
insert into PROCESS_SCHED
```

```
(process_name, task_name, task_cycle, task_mode, sched_start, task_intrvl, task_status, task_priority, slide_time, db_name, sql_query, debug_level, plat_id, usg_crt_hour, usg_plat_id, usg_version)
```

```
VALUES ('com01', 'com01', 'N', 0, SYSDATE, 60, 0, 2, 55, 'BILL', 'EXT_CONTACTS.ext_contact_id=122', 1, NULL, 0, NULL, 0);
```

Where,

- **Process_name:** Name of the process to run (com01, com02, com03 and so on.).
- **Task_name:** Usually it is the same as the process_name.
- **Task_cycle:** Recurring mode (R) or Non-recurring mode (N).
- **Task_mode:** Equal to 0 for production mode.
- **Sched_start:** Schedule time to start the process.
- **Task_intrvl:** Applicable only in recurring mode. The sleeping time between each run (Default Value = 60 seconds).

Command to Run

```
COM com01 <Unscaled server ID>
```

Unscaled server id = 4 if run on BLUS (CV mode)

Unscaled server id = 12 if run on ORP (CV/RT mode)

Expected Output and Location of Output

COM transfers the usage file to the location specified in EXT_CONTACTS.arbor_ready. For each usage file transferred, COM inserts a row into the FILE_STATUS and FILES_TO_PROCESS tables on the unscaled database. Therefore the downstream process (CMCAP) knows from where to take the usage file.

List of Success/Failure Criteria for COM Process

After complete processing, COM updates its status on PROCESS_STATUS table on unscaled database. Normally, PROCESS_STATUS.process_status = LOGOUT indicates the process run successful.



See table PROCESS_STATUS in the appropriate Comverse ONE Database Reference for information on complete process_status value.

The control report file should be the first file to look at after COM finish processing. It has the details as mentioned in the example below:

After COM finishes processing, first take a look at the control report file. It mentions the details as mentioned in the example below.

Example:

ARBOR COM CONTROL REPORT 20 Mar 2009
AUTHOR: cbs_owner 17:10:19
PROGRAM: COM PAGE: 1

Process: com08

Start time: 2009/03/20 17:10:19

CONTACT BYTES	SOURCE STATUS	SRC	ACCESS	SEND/ RECEIVE	FILE	SERVER	FILE NAME	
ID	ID	TYPE	METHOD		ID	ID		
=====	=====	=====	=====	=====	=====	=====	=====	
=====	=====					=====		
=====	=====							
128	1	usage	LOCAL	RECEIVE	2309	4	pqa_test.dat	640
SUCCESS								

128	CONNECTION STATUS: LOCAL			RECEIVE	1 good, 0 duplicates, 0 other errors			
SUCCESS								

End of Report

If something is wrong in the control report, then take a look at the log file to see what went wrong.

Example:

20090320 17:10:09.160 (I 0000000) [0][0][001] EMIT_BYPASS	Value of environment variable
OAM ENV CONN MA: 'FALSE'	

20090320 17:10:09.160 (I 0000000) [0][0][001] EMIT_BYPASS Value of environment variable
OAM_ENV_RECONN_ATTEMPT_NUMBER: '10'

20090320 17:10:09.161 (I 0000000) [0][0][001] EMIT_BYPASS Value of environment variable
OAM_ENV_RECONN_ATTEMPT_INTERWAIT: '30'

20090320 17:10:34.618 (I 4003138) [0][0][001] BAD_MODL_REDUCT NO REPORT
REDUCTION: Bad Module Level Factor=1/1

20090320 17:10:35.875 (I 4000697) [0][0][001] PUT_STATE_INFO Logging state TASK_INIT

20090320 17:10:37.813 (I 4000697) [0][0][001] PUT_STATE_INFO Logging state PROC

20090320 17:10:38.294 (I 4002726) [0][0][001] COM_BEGIN_SOURCE Begin accessing ext_
contact_id 128 (0)

20090320 17:10:40.217 (I 4002962) [0][0][001] COM_BEG_LOCAL_REC COM Begin LOCAL
RECEIVE - retrieve files

20090320 17:10:43.143 (I 4002868) [0][0][001] COM_GOOD_COUNT **1 files moved from data
directory; 1 success**

20090320 17:10:43.143 (I 4002870) [0][0][001] COM_BAD_COUNT 0 duplicates; 0 locked; 0 DB
errs; 0 other

20090320 17:10:43.143 (I 4002814) [0][0][001] COM_END_LOCAL End of LOCAL cycle

20090320 17:10:43.143 (I 4002740) [0][0][001] COM_END_SOURCE Finished accessing ext_
contact_id: 128 (0)

20090320 17:10:43.623 (I 4000697) [0][0][001] PUT_STATE_INFO Logging state POSTPROC

20090320 17:10:47.744 (I 4000697) [0][0][001] PUT_STATE_INFO Logging state LOGOUT

Troubleshooting Guidelines



See Chapter Communication (COM) in the *Comverse ONE Billing Technical Reference*.

Chapter 6

CSS Process

6

orded (sender
The desti
notifying
ng The noti
ieve The m
t access To



CSS Process Description

The different administration tools are described in the Self-Service Administration guide.



See Appendix A of the Self-Service Administration guide for detailed tool reference.

Prerequisites for Running CSS Process

The different processes or utilities are dependent of their respective run-time environment. Most of the utilities are installed and deployed in the Self-Service Platform RTE.



See Self-Service Installation Guide and Self-Service OCM Installation Guide for more details.

Other processes and utilities are hosted in an application server. These are configured as standard J2EE applications.



See appropriate installation guide for the application server installation and administration.

Running CSS Process

The details of utility command lines are described in the Appendix A of the Self-Service Administration Guide.

Environmental Variables

The RTE tools and commands do not depend on the environment variables. The environment is configured during the installation process and all the values are stored in the nmy.env.sh or nmy.env.cmd files in <home>/rte/bin/env.sh or <home>/rte/bin/env.cmd folders.

System Parameters

Self-Service processes and utilities do not depend on the system parameters. The configuration is stored in the property files. See respective documentation for more details.

Optimal CSS Process Schedule



See Self-Service Administration Guide for more details on:

Administration tasks: Chapter 3 Administrating Solutions

Monitoring tasks: Chapter 4 Monitoring Solutions

Auditing tasks: Chapter 5 Auditing Solutions

Maintenance Guidelines

This section only lists essential maintenance tasks.



See Self-Service Administration Guide and the Self-Service Operational Maintenance Guide for more details.

File System Maintenance

Web/App Server Tier

On the Web and Application Server tier, the maintenance monitors and administrates:

- Web Server logging files: This is not product specific. See Web Server documentation.
- App Server logging files: This is not product specific. See Web Server documentation.
- Self-Service logging files: See Administration Guide to configure logging files. It is recommended to use rolling file to control their size.
- WFS file system: Depending on the usage of this component the file system that hosts it must be monitored. This component is not used by default by the application. Its usage is project specific. See Administration Guide for more details.
- Catalog Media files: The size and growth of the Catalog Media file must be analyzed. The file system must be monitored.

Database Tier



See Database Maintenance documentation for details on any file system specific maintenance on the Database tier.

The Tablespace and the data files maintenance are discussed in the Database Maintenance section.

Integration Tier

On the Integration tier, connectors and processors are hosted in the RTE.

All run-time file activities are stored in <home>/rte/var. This file system hosts:

- Log files for all connectors and processors: See Administration Guide to configure logging files. It is recommended to use rolling file to control their size.
- Connector file queues: For retry and error queues. The total size must be monitored.
- WFS file system: For asynchronous report processing. The total size must be monitored.

Database Maintenance

CID Database

Analyzing statistics on the CID database must be run regularly. The actual frequency is dependent on the user activity.

The CID is a high volume OLTP database. The operational parameters of the database must be monitored by a DBA.

The sizing and growth of different objects must be analyzed and monitored by the project team. Tablespace must be sized depending on this analysis. It is important to effectively monitor free space to prevent capacity overflow.

The sizing and monitoring includes:

- Billing Entities: account, contract (subscriber), and contact
- User Entities: organization, member, user, and personal data
- Transactions: persistent action manager, request, trouble ticket, and approval process
- Invoices: main and sub invoices
- History: user event, and usage info

Both initial sizing and estimated growth must be carefully analyzed to plan the data file monitoring and administration.

OCM Database

Analyzing statistics on the OCM database must be run regularly. The actual frequency is dependent on the catalog activity.

The OCM database is not a high volume database. Once sizing is determined, regular parameters must be checked. No specific monitoring is needed.

CBU Database

This database is not deployed in Comverse ONE Self-Service solution.

Archiving Maintenance

WFS File System

When asynchronous reporting is activated, the saved reports must be purged at a project specific frequency.



See Administration Guide for more information on report admin tool.

Other uses of the WFS file system are project specific. The need for purging and archiving must be determined by the project team.

CID Database

The CID database must be purged and archived on a regular basis.

Purging operations include:

- Main and Sub invoice tables
- Request tables
- User event
- Shopping Cart (Persistent action manager)
- Jobs
- Notifications
- Approval

The actual frequency is dependent on the system activity.



See Administration Guide for the specific command to use.

The CID database must be archived.

OCM Database

There is no need of purging operation for the OCM database. OCM does not hold any transactional data that requires purging.

The OCM database must be archived.

CBU Database

This database is not deployed in Comverse ONE Self-Service solution.

Chapter 7

MIUB Process

7

orded (sender
The desti
notifying
ng The noti
ieve The m
cT access To



MIUB Process Description

MIUB is to repackage corrected records for reprocessing, after basic routing/segmenting of USAGE data to the appropriate database.

Prerequisites for Running MIUB Process

Run WIN-MIU before MIUB to do correction for the records. Make sure the host_contacts is configured correctly for MIUB against the running machine and also specify the output file locations.

Dependencies on Other Processes

1. The records should be released by WIN-MIU after correction. That is, WIN-MIU should be closed before MIUB.
2. If WIN-MIU is not run before MIUB, MIUB may do successful routing for some records, but later these records will fail in CCAP.

Running MIUB Process

1. Run from UPM.
2. Run from command line.

```
MIUB <process_name> <Unscaled server ID> <where_clause> PICK
<output_
file_id> <target_db_a id> (i.e. MIUB miub01 4)
```

Example:

```
MIUB miub01 4 (run CMCAP on Converge env)
```

```
MIUB miub01 12 (run CMCAP on Realtime env)
```

```
MIUB miub01 4 'where file_id=123' (use where_clause to reduce the
records to be processed)
```

```
MIUB miub01 4 PICK 124 9 (the file 124 with target_db_a=9 was created in
last run of MIUB, we can use this command line to reprocess the records
in file 124)
```

Process_name: process name to be run. miub01, miub02, and so on

Unscaled server ID: server ID of Unscaled or ORP DB

Where_clause:

PICK: this is used for disaster recovery, MIUB reprocesses the old MIUB files in PICK mode.

output_file_id: together with PICK, file ID of the MIUB files to be reprocessed

target_db_a_id: together with PICK, target_db_a of the MIUB files to be reprocessed

Configuration of MIUB Process

There are no configuration files for this process. All configurations are done in the database.

Environmental Variables

The following are the environmental variables that affect this process.

- ARBORDIR: Top level directory where the apps are installed
- ARBORLOG: Path where the log file must be placed (/staging/billing/log)
- ARBORCTRLRPT: Path where the control report file must be placed (/staging/billing/reports)
- ARBOR_CATALOG_DATABASE: Catalog database server (CTLG for Converge, MAIN for Real time mode)
- ARBOR_CATALOG_QUERY: Catalog database instance (CTLG for Converge, MAIN for Real time mode)
- ARBORDATA: Top level directory where the usage files are located (/staging/billing/data)
- ARBORBIN: Path where all the binaries are located
- ARBORDBU: User schema to log into the database (bs_owner)
- OAM_ENV_CONN_MA:
 - = TRUE (to connect to the security server to get database password)
 - = FALSE (Not to use security server for database password)

System Parameters

Table 4 System Parameters

Module	Parameter Name	Default	Use
MIUB	LOGLEVEL	MEDIUM	Configure modules to log errors of a designated severity level or higher. There are three levels of log detail: LOW — logs ERROR, ALARM, and FATAL messages MEDIUM — logs INFORM, WARNING, ALARM, ERROR, and FATAL messages HIGH — logs all messages and creates SQL logs
MIUB	TRA_SWITCH	0	Set to 1 to turn on detailed logging for any module. At this setting, SQL log files are written to the \$ARBORLOG directory and high-level module information is sent to the standard output stream (the terminal screen, unless the output is redirected).

Optimal MIUB Process Schedule

This process is run on demand.

Reviewing MIUB Process Activity

Log files are located under \$ARBORLOG directory. Control reports are located under \$ARBORCTRLRPT directory.

Expected Input for MIUB Process

The records in CDR_DATA_WORK are corrected initially by WIN-MIU, and then it is processed by MIUB.

Expected Output and Location of Output

For input records it can generate one output file per target rating DB. The new file is put into the specified directory (HOST_CONTACTS. HOST_OUTPUT). The relate records are inserted into usage_file_status and usage_files_to_process.

List of Success/Failure Criteria for MIUB Process

It is regarded as a successful run that the expected records are picked up by MIUB, and written into MIUB files. In detail, miu_disp_code is set to 2 and miu_disp_status is set to 0 for these successful records.

If some expected record is failed, then the miu_disp_code is set to 0 and miu_disp_status is set to 1.

An easy way is to check the successful record number in the log as below:

```
=====
20090327 04:38:17.134 (E 2003420) [0][0][001] MCAP_STAT01          Good Records: 1
20090327 04:38:17.134 (E 2003422) [0][0][001] MCAP_STAT02          Soft Error
Records:      0
20090327 04:38:17.134 (E 2003424) [0][0][001] MCAP_STAT03          Total Records:1
=====
```

Troubleshooting Guidelines

1. Error: Database login record initialization failed!
 - a. If environment OAM_ENV_CONN_MA=TRUE, then check the security server connectivity.
 - b. If environment OAM_ENV_CONN_MA=FALSE, then check if the encrypt password file exist or not. Use the binary envtopwe.exe to create the password file.
 - c. If a) and b) is fine, then check the environments below to see if they are correctly set:


```
ARBOR_CATALOG_DATABASE
ARBOR_CATALOG_QUERY
ARBORDBU
```
2. Error: MIUB ought to be run on admin DB

Check if the server ID in command line is unscaled server
3. Error: Could not find type_id_usg: 40012

Missing configuration on table AUT_INITIAL_REF for the corresponding initial_aut_id or the version of the initial_aut_id is out range of the trans date of the usage record.
4. Other errors

Turn on LOGLEVEL/TRI_SWITCH to trace more SQL logs.

Maintenance Guidelines

File System Maintenance

The location which is defined in HOST_CONTACTS for MIUB is checked for the availability.

Archiving Maintenance

Run UFA once per two weeks or once per month.



See Operational UFA Process chapter for more details on using UFA.

Chapter 8

ORLTP Process

8

orded (sender
The desti
notifying
ng The noti
ieve The m
cT access To

e
v

ORLTP Process Description

ORLTP:

- inserts errored Outage Records into the ORP DB.
- routes redirected usage.

ORLTP is responsible for updating and rolling up the audit trail on the ORP DB.

Prerequisites for Running ORLTP Process

The prerequisites are:

- Configuration of Environment variables must be correct.
- Configuration of a row in HOST_CONTACTS for the machine where this utility is running must be correct.
- Configuration of the database names in SERVER_DEFINITION.dsquery field in the ORP database must be correct. The database names must appear in the TNS file.

Dependencies on Other Processes

ORLTP is started by mshell. ORLTP requires UPA to run unless the OAM_ENV_CONN_MA is set to FALSE.

ORLTP processes error/redirection files generated by CMCAP and CCAP. ORLTP sleeps if there are no files for it to process.

Running ORLTP Process

Run from UPM

```
ORLTP <Process_Name> <ORP DB ID>
```

Example:

```
ORLTP or01 12 (ORP db id is 12)
```

```
ORLTP or03 12 (ORP db id is 12)
```

where,

- **Process_Name:** The process name to be run (01,02, or 03 and so on).
- **ORP DB ID:** The server ID of the ORP DB (Billing DB).

Configuration of ORLTP Process

\$TNS_ADMIN/tnsnames.ora must have appropriate entries for accessing the ORP DB. No other configuration file is needed. All configurations are done in the database.

Environmental Variables

- **ARBORDIR:** Top level directory where the apps are installed.
- **ARBORLOG:** The path where the log file must be placed (/staging/billing/log)
- **ARBORCTRLRPT:** The path where the control report file must be placed (/staging/billing/reports).

- **ARBOR_CATALOG_DATABASE**: Catalog Database.
- **ARBOR_CATALOG_QUERY**: Catalog Database Server.
- **ARBORDATA**: Top level directory where the usage files are located (/staging/billing/data).
- **ARBORBIN**: All binaries are located.
- **ARBORDBU**: User schema to log in database (that is, cbs_owner).
- **TNS_ADMIN**: show the position where tnsnames.ora locates in.
- **OAM_ENV_CONN_MA**
 = TRUE - To connect to the security server for getting the database password
 = FALSE - Not to use security server for getting the database password
- **TZ**: Determines the time zone where the process is running. In several places URC takes the current system time based on the time zone and inserts that into the database. The TZ setting for ORLTP should match the TZ setting of the Oracle instance. Configure it according to UNIX TZ variable configuration.
- **ORLTP AUTOEXIT**: Used in testing.
 = 0 and Negative numbers - Exits after processing these many files and finding no more processing to do.
 = Positive number (N) - Exits after N number of files are processed.

System Parameters

Table 5 System Parameter

Module	Parameter_Name	Default	Use
OLTP	LOGLEVEL	MEDIUM	<p>Configure modules to log errors of a designated severity level or higher. There are three levels of log detail:</p> <ul style="list-style-type: none"> ■ LOW - Logs ERROR, ALARM, and FATAL messages ■ MEDIUM - Logs INFORM, WARNING, ALARM, ERROR, and FATAL messages ■ HIGH <p>Logs all messages and creates SQL logs</p>
OLTP	TRA_SWITCH	0	<p>Set to 1 to turn on the detailed logging for any module. In this setting, the SQL log files are written to the \$ARBORLOG directory and high-level module information is sent to the standard output stream (the terminal screen, unless the output is redirected).</p>

Module	Parameter_Name	Default	Use
OLTP	REPORT_LEVEL	MEDIUM	Configure modules to log configuration information of a designated severity level or higher. There are three levels of log detail: <ul style="list-style-type: none"> ■ LOW - Logs ERROR, ALARM, and FATAL messages ■ MEDIUM - Logs INFORM, WARNING, ALARM, ERROR, and FATAL messages ■ HIGH - Logs all messages and creates SQL logs
OLTP	INPUT_QUEUE_SIZE	10	Number of files the LTP can have locked but not actively being worked on
OLTP	BATCH_INSERT_SIZE	1000	Number of records to insert in one DB operation.
OLTP	INSERT_NODES	1	Number of processing threads running in the LTP.
OLTP	SHUTDOWN_WAIT_SEC_ON_ERROR	30	Maximum delay before killing threads during shutdown.
OLTP	PROCESS_TRUNCATED_FILES	0	Enable processing of partial files that have been truncated.
OLTP	ROLL_UP_INTERVAL	5	Short delay (in seconds) between searches for audit trail records to rollup.
OLTP	LONG_ROLL_UP_INTERVAL	60	Long delay (in seconds) between searches for audit trail records to rollup.
OLTP	WAKE_UP_INTERVAL	5	Short delay (in seconds) between searches for files to process.
OLTP	LONG_WAKE_UP_INTERVAL	60	Long delay (in seconds) between searches for files to process.
OLTP	LTP_IGNORE_ERRORS	0	Ignores file size differences between actual file size and total_records recorded in USAGE_FILE_STATUS. Actual file size is used.
OLTP	MAX_RPT_FILE_ROLL	10	Maximum Number of report and log files per report after size is exceeded. Range 1-10,000
OLTP	MAX_RPT_FILE_SIZE_MB	10	Number of megabytes per report and log files before the file rolls. Range 1-1024 (1MB-1GB)
OLTP	MAX_HUNG_PROCESS_MIN	20	Number of minutes the process waits before attempting to self terminate due to a hung thread. Range 10-120

The **file input thread** uses WAKE_UP_INTERVAL and LONG_WAKE_UP_INTERVAL to determine how often to poll the USAGE_FILE_STATUS table for files to process. It uses WAKE_UP_INTERVAL and LONG_WAKE_UP_INTERVAL in the following way:

1. It polls USAGE_FILE_STATUS and keeps a count of the number of times in a row it has not found a file to process.

2. It waits according to the value of the count.
 - 0 - 9: It sleeps for one second.
 - 10 - 29: It sleeps for WAKE_UP_INTERVAL seconds.
 - >= 30: It sleeps for LONG_WAKE_UP_INTERVAL seconds.
3. Start again at 1.

When a file to process is found it resets the count back to 0.

In this version ORLTP, the sleeps are implemented as a succession of 1 second interval that check the shutdown requests each second, and thus a shutdown request is not delayed until the total sleep interval passes. This makes ORLTP more responsive to a shutdown request.

This input mechanism is a throttled wait mechanism, that is, when there is work to do the ORLTP polls often. If the amount of work slows down or stops for a while, ORLTP reduces its polling rate accordingly. As soon as an input file becomes available ORLTP resumes back to full speed processing.

HOST_CONTACTS

Each machine where the ORLTP is run requires a HOST_CONTACTS entry configured for that machine.

The HOST_CONTACTS entry for ORLTP can have a HOST_MODULE value of ORLTP or NULL.

The processes look first for host_module = ORLTP and then for NULL. Once a match of the machine IP address and the HOST_NAME is found then the searching ends. If the Same HOST_NAME appears in 2 rows with host_modules of ORLTP and NULL, the row with host_module = ORLTP is used.

HOST_CONTACTS rows must have same configuration in both ORP and UNSCALED databases.

The host_ready, host_done and host_output directories must be configured for ORLTP host_contacts rows.

Optimal ORLTP Process Schedule

ORLTP can run as a batch or a server process as indicated in the task_cycle field of entries in PROCESS_SCHEDULE table. By default, ORLTP runs as a batch process, that is, it runs, processes all current content, and then terminates itself.

Reviewing ORLTP Process Activity

The file and configuration report are located under \$ARBORLOG directory.

Example - Log File Name Convention

orltp01-090422-045516-09-1306862.BATCH

orltp01-090422-045516-09-1306862.cfg

The .BATCH file is the log. If the log file exceeds 10MB, then it is renamed to <log>.1 through <log>.10. The current log never has a number after .BATCH and the oldest log has .BATCH.10. After .10 is reached the log is deleted and only the most recent 10 logs are kept.

The configuration file is the configuration report, a report that contains the configuration that was read from the DATABASE when the process ran.

The Control report of the process activity is located under \$ ARBORCTRLRPT directory.

Example: Log File Name Convention

orltp01-090422-051355-09-1146966.rpt

IF the ARBORCTRLRPT directory is not specified, then the report is placed in the \$ARBORLOG directory. The process activity is reported in both the log and the Control report.

A status report of the process can be created by sending a kill -USR2 <pid>. The status report reports the recent activity and the current thread status. This report ends with .status and is written to the \$ARBORLOG file.

The configuration reports are renamed daily -<process_name>_LASTRUN_DAY_X.extension where,

X is number of the days of the week the logs were started.

The renaming occurs every day (once) and during normal process termination so when everything is operating normally logs for the last 7 days are available. The configuration and reports are not renamed and are kept for 30 days in situations where the process fails for some reason.

Expected Input for ORLTP Process

Before running OLTP, it is expected that it is scheduled to run by inserting the following SQL to PROCESS_SCHED table into the unscaled database:

```
insert into PROCESS_SCHED
(process_name, task_name, task_cycle, task_mode, sched_start, task_intrvl,
task_status,
task_priority, slide_time, db_name, sql_query, debug_level, plat_id, usg_
crt_hour, usg_plat_id, usg_version)
VALUES ('orltp01', 'or01', 'N', 0, SYSDATE, 0, 0, 1, 55, 'BILL',
NULL, 1, NULL, 0, NULL, 0);
```

where,

- **process_name:** Name of the process to be run (orltp01, orltp 02, orltp 03, and so on.).
- **task_name:** Usually it is same as the process_name but orltp01 is too long so a shorter name is used here.
- **task_cycle:** The task cycle.
 - Recurring mode (R)
 - Non-recurring mode (N)

N is used to run ORLTP as a batch process (that is, process all input and terminate). If you want to run ORLTP continuously, then set the task_cycle to R.
- **task_status:** The status of the task.
 - 0 - new
 - 1 - started
 - 2 - completed

The value must be 0 for new tasks.
- All other fields: Ignored

If there are no PROCESS_SCHED rows, then OLTP runs in Non-recurring mode.

ORLTP only processes the OR error file from CCAP and CMCAP, including liability redirection file from CCAP.

Expected Output and Location of Output

ORLTP creates different output when dealing with different source file.

Table 6 Output and Location of Output

Data Source	Output	Location
Error usage records from offline processing	Data inserted into the BLUS DB	Data is inserted into: <ul style="list-style-type: none"> ■ CDR_DATA_WORK ■ CDR_DATA_WORK_TAX
Redirection records	Inserted into a outbound data file (file_class = 140)	HOST_CONTACTS.host_output directory
Processed input files	USAGE_FILE_STATUS with a file_status in (4,10)	HOST_CONTACTS.host_done

The Output files from ORLTP ALWAYS have the USAGE_FILE_STATUS.link_id field set. The purpose of this field is to link the output file to the original input file by setting the link_id field of both rows to the same value.

List of Success/Failure Criteria for ORLTP Process

A file is considered to be processed successfully by ORLTP if the records in the file are inserted into the ORP DB or output to the LR file. Otherwise the processing of the file is considered as failed.

Check the USAGE_FILE_STATUS.file_status on ORP DB. If file_status turn to 4 or 10 and link_id = 0, then the process is successful. If the Link_id values are not equal to NULL, then it indicates that some records were either redirected or have encountered errors and a new output file containing the data is created.

Troubleshooting Guidelines

The Transfer errors are reported in the UFS (USAGE_FILE_STAUTS on ORP DB).

UFS.file_status (3 ftp errors or 6 general errors)

If the transfer data is sent to an error file due to parsing problem, the entry in UFS has always a link_id and a new error output file with file_class 520,530, or 540.

Start with the LOG and the CFG files. There is a clear statement about the problem, any alarms raised, and the configuration setting while running the process.

Two main categories of errors are:

- Startup

The log file reports any configuration problems that prevent the operation. Check the log file to know why the process failed.

- Operational Problems

The Operational problems that identify a problem with the system during runtime appear in the process logs.

If ORLTP fails due to an unrecoverable error during operation, the quick means to locate that error is to search **SignalS** in the log. The cause of the error appears immediately above the message with **SignalShutdown** in the message.

Maintenance Guidelines

File System Maintenance

The file system that needs to be checked is HOST_CONTACTS.host_done directory where the module is ORLTP. The usage files placed in this directory are the files that are successfully processed. The files under this directory can be removed safely if needed.

The UFP process automatically ages out and removes the processed data files found in HOST_DONE. Similarly, UFP automatically ages out the LTP output files.

Database Maintenance

A utility module name UFA is used to archive or purge the following tables:

- FILE_STATUS
- USAGE_FILE_STATUS
- USAGE_FILE_STATUS_HISTORY
- USAGE_FILE_COUNTS
- USAGE_FILE_COUNTS_HISTORY
- USAGE_LOCAL_COUNTS

Archiving Maintenance

Run UFA once in two weeks or once in a month.



See UFA Process chapter for more details on using UFA.

9

Chapter 9

Product Catalog Propagation Process

orded (sender
The desti
notifying
ng The noti
ieve The m
cT access To

e

v

Propagation Process Description

Propagation is the process of installing the data in a PC Version into a Comverse ONE, such that the data becomes available for rating and billing. The data provisioned via the PC GUI is stored in the offline database (PCAT) and propagation helps to export/install the data in PCAT to the (selective) target databases group as a target group of a particular type of system (Production/Test).



PC SYNCH (with PURGE) and SYNCH should not be done on Live sites without prior consent from Tier4 or Comverse One Product R&D. These are critical operations. If these operations are not performed correctly, then it can lead to downtime.

Prerequisites for Running Propagation Process

UPM/Service Side:

1. UPM should be installed.
2. Workpoint should be installed.
3. Propagation process flow should be installed (includes IdCheck).
 - a. Make sure the following command is run to load the process models:


```
>cd $JBOSS_HOME/cbs_workflow/bin
>./LoadProcessModels.sh -f ../process_models/wppm_propagation.zip
-u wfadmin -p wfadmin123
```
 - b. Make sure the following command is run to clean up ID conflict issue.


```
>cd $JBOSS_HOME/cbs_workflow/bin
>./IdCheckCmd.sh FIX NOQUITE
```
4. Security Server has the username/password for the offline pc (pcat).
 - a. If you are using cbs_owner/comverse as the username/pw for pcat, UPM sets the account for you.
 - b. If you have your own username/password, you need to set it yourself.


```
>mshell secadmin/password
>list_credential
```

If you cannot find the username/passwd for your pcat, then start from step #4.

```
>store_credential -type database -uid xujo01_ga31_tr1 -dbtype UPC -in CBS3009 -
pwd arbor123
```

PC (Client) Side:

1. Java version 1.6 JRE/JDK.
2. There is no space in the installation path.
3. The following files are required under <user dir>\pcSettings\ folder:
 - ☐ workpoint-client.properties
 - ☐ jdbc.properties
 - ☐ security.properties
4. Check the configuration of the target databases.

- a. Check if the target DB is defined correctly in the DP_TARGET_DB tables. Remember that all the URL links of the DBs are setup/point to the correct IP.
 - b. Check if the URL of the Webservice is also set up for OCM.
 - c. Check if the Target group is defined correctly in the DP_TARGET_GROUP_REF/VALUES. Double check the target system - PRODUCTION or TEST. If you expect an update of propagation status in after propagation, then choose PRODUCTION.
 - d. Use tracepath to check if the network connection speed between the UPM and the Target DB. Compare few (if not all) results of the target DBs and check if there are DBs that have abnormal speed. Assume for UPM - each Target DBs should be < 1ms
5. Check if Port 80 is opened for Workpoint XML Servlet connection.
 6. Check if the Versions are in proper state.
 - a. All Reseller Version(s) of the PC Version that has to be propagated needs to be checked in.
 - b. Service Version of the PC Version that has to be propagated also needs to be checked in.
 - c. PC Version promoted (DDD and Tutorial)
 7. Table locked
 In a special conditions (incorrect DB operation, upgrade interrupted, and so on), some of the PCAT tables may be locked (especially need to check MASTER_PROPAGATE).

Running Propagation Process



See

<http://us-dnv-sp01/sites/SQPM/PC/PC%20Knowledge%20Transfer/Propagation/PCPropagationTur.ppt>.

Summary of the steps given in the PPT slides:

1. Check in Versions (Reseller/Service/PC).
2. Go to the Propagation Dialog.
3. Select **Propagation Target** (Group).
4. Select Propagation Type (Major/Minor/None version propagation).
5. Set the start time.
6. Select/Unselect the **Target Database** (Optional).
7. Click **Save**.
8. Check Propagation status.

Configuration of Propagation Process

1. Check the \$ORACLE_HOME/network/admin/tnsnames.ora.
 Make sure the pcat is pointing to the offline pc that is using the propagation.
2. \$.JBoss/cbs_workflow/propagation.properties (the word in red need to match the actual value)
 Check the following lines to make sure it is pointing to the pcat.

- ```
upc.user=cbs_owner
upc.pass=comverse
upc.db=pcat
upc.jdbc.url=jdbc:oracle:thin:@<pcat ip>:1521:pcat
```
3. \$JBOSS/cbs\_workflow/ propagate-ksh.properties (the word in red need to match the actual value)
- Check the following lines to make sure the env var is correct (check the words marked in red):
- ```
ORACLE_HOME=/oracle/product/10.2.0
DBMS=oracle
PRODUCT_CATALOG_DATABASE=pcat
ARBORBIN=/home/jboss/cbs_workflow/propagation
LOGDIR=/home/jboss/cbs_workflow/propagation/LOG
PATH=/bin:/usr/bin:/usr/sbin:/oracle/product/10.2.0/bin:/home/jboss/bin
```

Environmental Variables

Before the installation of the Workflow-propagation, set the environment variable (replace the words marked red to the correct value of the current system) on UPM (the words marked in red need to match the actual value):

```
export PRODUCT_CATALOG_DATABASE_USER=cbs_owner
export PRODUCT_CATALOG_DATABASE_PASS=comverse
export PRODUCT_CATALOG_DATABASE_HOST=10.8.32.53
export PRODUCT_CATALOG_DATABASE_PORT=1521
export PRODUCT_CATALOG_DATABASE=pcat
```

Optimal Propagation Process Schedule

It is better to run the Propagation Process during off-peak period of the network traffic.

Reviewing Propagation Process Activity

\$JBOSS_HOME/cbs_workflow/propagation/LOG/

The log file name format is:

propagate.DDMMYY.<Work_Id>.<Database_ID>.<Time Stamp>.log

Example:

propagate.11May09.617.9.3546.log

Expected Input for Propagation Process

The PC Version to be propagated is promoted to Validated state in the database table before Propagation of the PC Version can be selected/propagated via the PC GUI.

Expected Output and Location of Output

Check the PC Version data in the online target databases. For all selected target database from the **Propagation** dialog, the propagated PC version's data, including the data from reseller versions and service version, are shown in the database tables. The PC Versions status is set depending on the propagated PC Version active date.



See <http://us-dnv-sp01/sites/SQPM/PC/PC%20Knowledge%20Transfer/Propagation/PCPropagationTur.ppt>

List of Success/Failure Criteria for Propagation Process

Success:

1. Check Propagation Status (Versioning > PC Version > Check Propagation Status). The parent Propagation Work ID is shown as **Completed** or **Completed (non-blocking failed)**.
2. Check the PC Version tables in the Target DB and check if the current pc/reseller/service versions status is updated, active/inactive data is changed, based on the propagated versions active date.

Failure:

1. Check Propagation Status (Versioning > PC Version > Check Propagation Status). The parent Propagation Work ID is shown as Failed (non-versioned data propagation), Failed (blocking system), Failed and Failed (propagation request).
2. Check Propagation Status. If the status is in **InProgress** or **Committed** for a long time and you find error while running the following script on UPM:

```
> cd $JBOSS_HOME/cbs_workflow/propagation/LOG  
> grep ERROR *.<work_id>*.log
```



See <http://us-dnv-sp01/sites/SQPM/PC/PC%20Knowledge%20Transfer/Propagation/PCPropagationTur.ppt>

Troubleshooting Guideline



See <http://us-dnv-sp01/sites/SQPM/PC/PC%20Knowledge%20Transfer/Propagation/PC GUI and PC Propagation Verification Checklist.doc>

Maintenance Guidelines

File System Maintenance

1. Check the size of /home since LOG file is in:
/home/jboss/cbs_workflow/propagation/LOG
2. Make sure that there is space in /home. The following is an example where the propagation fails because of no sufficient space.

```
[root@upm1 LOG]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda6        2.8G  2.2G  374M  86% /
tmpfs            2.0G    0  2.0G   0% /dev/shm
/dev/sda7        3.6G  3.6G    0 100% /home
/dev/sda5        21G   16G  3.8G  81% /oracle
```

Database Maintenance

1. Make sure that the tables in PCAT are unlocked before/during propagation, especially the MATER_PROPAGATION table.
2. Make sure all the target database tables are unlocked before/during propagation.
3. Make sure the network connection is sound and fast before/during propagation.

Archiving Maintenance

Back up the workflow

```
cd $JBASS_HOME
tar cvf cbs_workflow_mmddyyyy.tar cbs_workflow
mv cbs_workflow_mmddyyyy.tar ~/wf_pkg
```


Chapter 10

RCS Process

10

orded (sender
The desti
notifying
ng The noti
ieve The m
cT access To

e

RCS Process Description

RCS is used to charge subscriber or account level RC/NRC (periodic/onetime) charges and create charge records for the charges.

Prerequisites for Running RCS Process

The prerequisite for running RCS process in either OnDemand or Batch mode is that TSP must be up and running.

Dependencies on Other Processes

Before running the RCS, start the TSP. Once TSP is up and running, RCS process is started. The SHM number of RCS and TSP should be same.

Running RCS Process

Running from UP

The steps to start OnDemand RCS are: (Batch RCS can be started with task mode = 0).

1. Log in as **root** in UNIX box.
2. Log in as **secadmin/password** into the mshell.
3. Start TSP: > `execute_process -g application -p tsprcs01 -d start -t i -bp tsprcs.`
4. Start URE/RCS: > `execute_process -g application -pr rcs01 -d start -t I -bp RCS -r "task_name=rcs01, server_id=3, task_mode=1".`
5. Shutdown RCS: > `execute_process -g application -pr rcs01_3 -d shutdown.`



See Appendix B, Unified Platform Management Shell Commands chapter in Unified Reference Guide for more information on the commands to start TSP and On demand RCS.

Running in Test Environments without UP

Run TSP in test environment:

1. Open a UNIX box, log in as **cbsuser**.
2. check whether TSP is running or not.
> `ps -ef | grep tsp_out.$SHM | grep -v grep`
3. Go to `$ARBORDIR/bin` and start TSP.
> `nohup ./tsp -nonomni -start -dump -db sdp99_n1.world -config $ARBORDIR/conf/tspCfg.xml -log tsp_out.$SHM &`

Run RCS in test environment:

1. Change shell to ksh. Source `envSetting.env`.
> `./envSetting.env`

2. To start the URE/RCS with batch mode to do a charge, run run_rcs with the account number as shown below:

```
. ./run_rcs <process_name> <account_no> <cleanup_account> <login_server_id>
```

RC_TERM_INST/NRC_TERM_INST Processing Status

The RC_TERM_INST and NRC_TERM_INST processing status are updated based on business logic flows. There are scenarios where a charge is not applied due to business level denials, configurations and so on. The list of all possible processing status is mentioned in the table below.

Table 7 RC_TERM_INST/NRC_TERM_INST Possible Processing Status

SL NO	Status	Description
0	PS_new	New (never processed)
1	PS_issued	RC/NRC issued
2	PS_insufficient_bal	Not issued, Insufficient Funds
3	PS_fraud_lock	Not issued, Fraud Lock
4	PS_suspended	Not issued, Suspended (and the RC is not the cause of the suspension)
5	PS_post_active_state	Not issued, Customer in Post-Active state
6	PS_writeoff_insuffic_bal	RC/NRC written off for insufficient balance
7	PS_writeoff_sub_acct_state	RC/NRC written off for subscriber/account state
8	PS_terminated	Terminated/Disconnected
9	PS_state_err	Not issued, invalid subscriber state
10	PS_fatal	Fatal Error
11	PS_issued_no_balance	Issued without balance. Postpaid write-off.

Configuration of RCS Process

The following two configuration files are provided for RCS and are shared with URE processes:

- ureCfg.xml
- pps.CoreUREModules.202

Environmental Variables

The following environment variables are set by envSetting.env file which is delivered as part of RCS package.

- RCS_IGNORE_ONDEMAND_GFRFATAL (Num)
 - When set > 0, bypass the gfr fatal error on on_demand mode
- ARBOR_HOME - Location where the package is installed
- ARBORDATA (Path) - Location path for log files
- RCS_BYPASS_FAILURE_ONDEMAND (Num)
 - When set = 1, ignores the failure on on_demand mode

System Parameters

Table 8 System Parameters

System Parameters Name	Module name	Possible Value
LOGLEVEL	RCS	= 0 = 1(LOW) = 2(MED) = 3(HIGH)
TRA_SWITCH	RCS	= 0(LOW) = 1(MED) = 3(HIGH)
RC_ROUND_METHOD	RC	= 0 = 1 = 2 = 3 = 4
DEPLOYMENT_MODE	CBS	= 1(PREPAID) = 3(CONV)
RETRY_LIMIT	RC	Num
SPLIT_RCS_FOR_RATE_CHANGES	RC	= 0 = 1
RING_BUFFER_SIZE	RCS	Num
INSTANCE_COUNT	RCS	Num
COMMIT_SIZE	RCS	Num

■ LOGLEVEL

- When set = 0, no debug information during run-time
- When set = 1, low debug level
- When set = 2, medium debug level
- When set = 3, high debug level

■ TRA_SWITCH

Set to 1 to turn on detailed logging for any module. In this setting, high-level module information is sent to the standard output stream (the terminal screen, unless the output is redirected).

TRA_SWITCH controls the log level; the actual log level does not go under the level of TRA_SWITCH.

When LOGLEVEL < TRA_SWITCH, then set the actual log level as TRA_SWITCH.

■ RC_ROUND_METHOD (RC)

Defines the rounding behavior for the prorated recurring charges and refunds (disconnect credits).

Set it as follows:

- 0 - No Rounding (Truncate)

- ❑ 1 - Round Down
- ❑ 2 - Round Up
- ❑ 3 - Round Nearest (exact 0.5 rounds down)
- ❑ 4 - Round Nearest (exact 0.5 rounds up)

The default value is 0 (No Rounding/Truncate) if the RC_ROUND_METHOD (RC) system parameter is not configured.

- **DEPLOYMENT_MODE**
 - ❑ When set = 1, RCS runs on prepaid deployment mode .
 - ❑ When set = 3, RCS runs on converged deployment mode.
- **RETRY_LIMIT**

When RC/NRC charge fails due to insufficient balance, set RETRY_LIMIT to decide on the number of times the RC/NRC has to be recharged. This system parameter has the same meaning with OFFER_RC_TERM_MAP.recovery_duration or BUNDLE_RC_TERM_MAP.recovery_duration, RCS reads the system parameter for default, and uses the value in the table.
- **SPLIT_RCS_FOR_RATE_CHANGES**

Set to 1 for BIP to split recurring charges into segment if a rate change occurs in the middle of a bill period. This enables BIP to calculate discounts and disconnect credits based on the actual pre-day contribution to the charge. The result is more accurate discount and credit totals on charges that have a rate or contract change in the middle of a bill period.

Set 0 for BIP to calculate recurring charges normally.
- **RING_BUFFER_SIZE**

The number of RCS loads RC/NRC instance size one time with ring-buffer mechanism.
- **INSTANCE_COUNT**

This number is used to control the max instance numbers which are handled at the same time by RCS.



NOTE

The system is shipped with optimal or default values (for the system parameters) and should not be changed without R&D recommendation.

Optimal RCS Process Schedule

Server Mode RCS

To start server mode RCS, execute the following command:

```
execute_process -g application -pr rcs01 -d start -t I -bp RCS -r "task_
name=rcs01, server_id=3, task_mode=1"
```

Batch RCS



SEE

See Cross Node Workflow section in Unified Reference Guide for information on nightly workflows for Batch RCS.

Reviewing RCS Process Activity

RCS log file path is defined in /staging/billing/conf /ureCfg.xml:

```
<LogFilePath><RCS log file path></LogFilePath>
```

Use ure boot argument -log to specify log name as shown below:

```
ure -log <log name>
```

Expected Input for RCS Process

For starting RCS, input these arguments:

RatingDBID:Rating DB server ID (RT mode, FOR CV mode this input does not matter)

BillingDBID:Billing DB server ID (CV mode, FOR RT mode this input does not matter)

ProcessName:RCS process name

```
ure -R -ratingdb $RatingDBID -billingdb $ BillingDBID -rcsname $ProcessName
```

Use ureCfg.xml file to specify other arguments.

Example: ureCfg.xml file

```
<ure>
  <ALL>
    <ARBOR_CATALOG_DATABASE>CAT</ARBOR_CATALOG_DATABASE>
    <AdminDbName>admin</AdminDbName>
    <AdminDbType>4</AdminDbType>
    <BILLING_DB_INSTANCE>CUST1</BILLING_DB_INSTANCE>
    <BILLING_DB_TYPE>BILLING</BILLING_DB_TYPE>
    <BILLING_DB_USER_NAME>cbs_owner</BILLING_DB_USER_NAME>

    <CcDbRemoteDisconnectIntervalInSec>300</CcDbRemoteDisconnectIntervalInSec>
    <CcDbUserName>cbs_owner</CcDbUserName>
    <DBLINK_CHECK_INTERVAL>60</DBLINK_CHECK_INTERVAL>
    <DBLINK_RETRY_INTERVAL>60</DBLINK_RETRY_INTERVAL>
    <DBLINK_RETRY_TIME>3</DBLINK_RETRY_TIME>
    <DbTransFailedAlarmLimit>2</DbTransFailedAlarmLimit>
    <DbTransFailedAlarmTimeLimit>100</DbTransFailedAlarmTimeLimit>
    <DebugTraceEnabled>false</DebugTraceEnabled>
    <DebugTraceTimestampMode>1</DebugTraceTimestampMode>
    <DefaultBillingDbId>3</DefaultBillingDbId>
    <DefaultCcDbId>999</DefaultCcDbId>
    <DefaultCcDbInstance>CCDB</DefaultCcDbInstance>
    <DefaultCcDbType>RATING</DefaultCcDbType>
    <DefaultOamDbId>555</DefaultOamDbId>
    <DefaultOamDbInstance>XE</DefaultOamDbInstance>
```

```
<DefaultOamDbType>UPM</DefaultOamDbType>
<DiscardAlarmCount>5</DiscardAlarmCount>
<DiscardAlarmInterval>300</DiscardAlarmInterval>
<HighResponseTimeAlarmLimit>10</HighResponseTimeAlarmLimit>
<HighResponseTimeAlarmTimeLimit>30</HighResponseTimeAlarmTimeLimit>
<InhibitCallHistories>0</InhibitCallHistories>

<InhibitZeroDurationCallHistories>0</InhibitZeroDurationCallHistories>
  <LogFilePath>/home/omni/Logs</LogFilePath>
  <MaxLogFileSize>1048576</MaxLogFileSize>
  <MaxNumRetries>10</MaxNumRetries>
  <MaxNumberLogFile>10</MaxNumberLogFile>
  <MinRetryInterval>30</MinRetryInterval>
  <MsgLifeTime>3000</MsgLifeTime>
  <NumConnectionsPerCcDb>6</NumConnectionsPerCcDb>
  <NumConnectionsPerOamDb>1</NumConnectionsPerOamDb>
  <NumConnectionsPerOtherSDP>6</NumConnectionsPerOtherSDP>
  <OamDbUserName>upmuser</OamDbUserName>
  <OptionForHung>KEEP</OptionForHung>
  <QueryFailureLimit>10</QueryFailureLimit>
  <QueryFailureTimeLimit>30</QueryFailureTimeLimit>
  <QueryRejectedAlarmLimit>10</QueryRejectedAlarmLimit>
  <QueryRejectedAlarmTimeLimit>30</QueryRejectedAlarmTimeLimit>
  <QueueCheckInterval>50</QueueCheckInterval>
  <RCS_CONFERR_THRESH_PERCENTAGE>80</RCS_CONFERR_THRESH_PERCENTAGE>
  <RCS_DBERR_THRESH_PERCENTAGE>80</RCS_DBERR_THRESH_PERCENTAGE>
  <RCS_MINIMUM_ACCU_COUNT>1000</RCS_MINIMUM_ACCU_COUNT>
  <RCS_WARNING_THRESH_PERCENTAGE>50</RCS_WARNING_THRESH_PERCENTAGE>
  <RECHARGE_AGENT_PORT>18001</RECHARGE_AGENT_PORT>
  <RcsDbLinkName>RCSLINK</RcsDbLinkName>
  <RetryMultiplier>31</RetryMultiplier>
  <RevAssTestMode>0</RevAssTestMode>
  <RevAssUssdTestMode>0</RevAssUssdTestMode>
  <SemWait>300</SemWait>
  <StatsCollection>1</StatsCollection>
  <TaxationEnabled>1</TaxationEnabled>
  <ThreadTimeout>120</ThreadTimeout>
  <UpdateFailureLimit>2</UpdateFailureLimit>
  <UpdateFailureTimeLimit>30</UpdateFailureTimeLimit>
```

```
<UpdateRejectedAlarmLimit>10</UpdateRejectedAlarmLimit>
<UpdateRejectedAlarmTimeLimit>30</UpdateRejectedAlarmTimeLimit>
<UreQHardLimitDelta>200</UreQHardLimitDelta>
<cancel>0</cancel>
<delta_conn>15</delta_conn>
<delta_init>30</delta_init>
<delta_trans>30</delta_trans>
<max_rejects>5</max_rejects>
<measbatchsize>50</measbatchsize>
<nolwp>0</nolwp>
<nomeas>0</nomeas>
<nosignal>0</nosignal>
<poll>45</poll>
<retry_int>300</retry_int>
<retry_max>100000</retry_max>
<sdp_rech_agent_port>18001</sdp_rech_agent_port>
<sdp_rech_agent_telnet_port>18002</sdp_rech_agent_telnet_port>
<sdp_rech_port>9999</sdp_rech_port>
<sdp_rech_timeout>5</sdp_rech_timeout>
<timeout>0</timeout>
<trans_retry>0</trans_retry>
</ALL>
<URE_O insid="0" dbid="0">
  <RcsAccountList></RcsAccountList>
  <RcsMeasurementIntervalInHour>24</RcsMeasurementIntervalInHour>
  <RcsOverrideAccountList>0</RcsOverrideAccountList>
  <RcsUseDbLink>1</RcsUseDbLink>
</URE_O>
<URE_Q insid="0" dbid="0">
  <delta_trans>5</delta_trans>
</URE_Q>
<URE_U insid="0" dbid="0">
  <BYPASS_FRAUD_FOR_INVALID_RECHG LENG>1</BYPASS_FRAUD_FOR_INVALID_
RECHG LENG>
  <EXTERNAL_RECHARGE_MAX_CARD_LENGTH>14</EXTERNAL_RECHARGE_MAX_CARD_
LENGTH>
  <EXTERNAL_RECHARGE_MIN_CARD_LENGTH>12</EXTERNAL_RECHARGE_MIN_CARD_
LENGTH>
  <SERVICE_VERSIONS_FOR_SDS>5</SERVICE_VERSIONS_FOR_SDS>
</URE_U>
```

</ure>

List of Success/Failure Criteria for RCS Process

To validate whether RCS ran successfully on batch mode, use the following Queries:

- Basic Query that finds everything that did not generate a charge, legitimately or not:


```
select rc_term_inst_id from RC_TERM_INST where ((next_apply_dt is not null and next_apply_dt < trunc(sysdate)) or ((next_apply_dt IS NULL) and (rc_term_inst_active_dt < trunc(sysdate))) and no_bill = 0 and status in (2, 3);
```
- Query that attempts to analyze why the instances are not processed. This is the workhorse.


```
select status, processing_status, process_error_code, count(*) from RC_TERM_INST where ((next_apply_dt is not null and next_apply_dt < trunc(sysdate)) or ((next_apply_dt IS NULL) and (rc_term_inst_active_dt < trunc(sysdate))) and no_bill = 0 and status in (2, 3), group by status, processing_status, process_error_code.
```
- Query that only selects the instances that should have been processed and were missed:


```
select rc_term_inst_id, next_process_dt, last_charged_dt, last_apply_dt, processing_status, process_error_code from RC_TERM_INST where ((next_apply_dt is not null and next_apply_dt < trunc(sysdate)) or ((next_apply_dt IS NULL) and (rc_term_inst_active_dt < trunc(sysdate))) and ((last_processed_dt is NULL) or (last_processed_dt < trunc(sysdate))) and no_bill = 0 and status in (2, 3);
```

Troubleshooting Guidelines



See Cross Node Workflow section in Unified Reference Guide for information on the sequence of RCS job execution in nightly jobs workflow.

To analyze RCS processing errors, use the following steps:

1. Make a list of all accounts that should have had RCS processed but were not.
2. Search RC_TERM_INST rows for each account. Check if the following conditions are met:
 - a. `rc_term_inst_start_dt < sysdate`
 - b. `(status = 2 and rc_term_inst_inactive_dt IS NULL) or (status = 3 and rc_term_inst_inactive_dt IS NOT NULL and processing_status != 8)`
 - c. `no_bill = 0`

Count the number of RC instances per subscriber which meet these conditions and compare them to the number expected per subscriber. Any mismatch indicates a provisioning or a workflow error. These errors have to be corrected through CSM.

3. For each subscriber, list the `rc_term_inst_id` values of every RC instance that satisfy the conditions mentioned in step 2. These are the instances that the RCS should have processed.
4. Open the Batch RCS log in the editor of your choice. For each instance in the process list, `searchinst_id: [rc_term_inst_id from list]`. The space after the colon is required.
5. If an instance ID is displayed, then it means that the RC instance was fetched from the database. Check the value of `last_processed_dt`. If this date is before the `sysdate`, then the

RCS has read the instance but it did not process it. This happens due to the following reasons:

- a. The instance data in the DB is corrupted and it failed validation. The failure is listed in the Log file.
 - b. The instance was validated by RCS but later it was not processed. This indicates a serious code error and a PR must be filed. Include the entire log file (not just an excerpt).
6. If the instance ID is displayed and last_processed_dt is same as the sysdate, then RCS has processed the instance and it has rejected it. This can happen due to many reasons and the specific reason is listed in the Log file. The most common reasons are insufficient balance, subscriber state suppressed issuing the charges, misconfigured PC data, and corrupted data in the instance record. Check the reason from the Log file and fix it. Note that some outcomes are considered valid results.
7. If the rc_term_inst_id value is not present in the log, then RCS has not fetched the instance. This is due to the inconsistent data in the DB.

Steps specific to Converged Deployment Mode

Check if:

- a. a row exists for the account in CMF with status ≥ -1 .
- b. a row exists for the subscriber in SUBSCRIBER_VIEW with view_status = 2.
- c. a row exists for the subscriber in SUBSCRIBER_BILLING with active dates that includes the system date.
- d. a row exists for the subscriber in SUBSCRIBER_KEY.
- e. a row exists for the subscriber in SUBSCRIBER_STATUS with active dates that includes the system date.
- f. a row exists in OFFER_INST_VIEW for the offer_inst_id with view_status = 2 for offer RCs.
- g. a row exists in OFFER_INST_KEY for the offer_inst_id for offer RCs.
- h. a row exists in BUNDLE_INST_VIEW for the bundle_inst_id with view_status = 2 for bundle RCs.

Steps common to Converged and Prepaid Deployment Mode

Check if:

- a. a row exists in OFFER_INST for offer_inst_id for offer RCs.
- b. a row exists in ACCOUNT_SUBSCRIBER for subscriber/account.
- c. a row exists in BUNDLE_INST for bundle_inst_id for bundle RCs.

Error Scenarios and Corrective Actions

Error Scenario

RCS cannot be started and complains about TSP error

Corrective Action

Check the status of TSP. If there are any table loading errors, then correct the contents of those tables. Restart TSP and then restart RCS.

Error Scenario

Module Loading Errors

Corrective Action

Check if the directory of the library in pps.CoreUREModules.202 is correct.

Error Scenario

Dblink/Connection Errors

Corrective Action

Check if the SERVER_DEFINITION on all test DBs has the correct DB names for each type. Ensure that the table does not have DBs listed that are not used. Perform the same check for tnsnames.ora.

Error Scenario

Changes to PCAT is not reflected while running Batch RCS

Corrective Action

If any new PC version is not submitted during propagation, then TSP should be rebooted.

Error Scenario

ORA-1013 error

Corrective Action

Increase the value of delta_trans (in ureCfg.xml) to 1800.

Prostack Trace

When an unknown error occurs use the following commands for obtaining prostack trace with process ID:

Dbx ure core - for a crashed process

Dbx <pid> ure - for a running process

Determining Duplicate Awards

RCS awards for a particular RC term and offer are configured in OFFER_RC_AWARD_MAP.

Check the award configuration to determine award value and unit type.

Check if the BALANCE_GRANTS table has awards with same creation date and same grant source for a single charge.

GRANT_SOURCE = RCS+<RC_TERM_INST_ID>+<OFFER_RC_AWARD_MAP_ID>

If there are multiple rows with same balance, grant source, and same creation date, then check if multiple charges are present on that day for that RC_TERM_INST_ID. There is only one grant per balance for one grant source per charge cycle.

Error Codes

The list of processing error codes is mentioned in the table below. The field is updated on RC_TERM_INST and NRC_TERM_INST when some technical failure happens in RCS which causes failure to charge for example, Incorrect configuration, record locking failure, db error and so on.

Table 9 Error Codes

Error Code	Error Code Name
104300	RCS_RATING_KEY_ERR
104301	RCS_RATING_DATE_ERR
104302	RCS_RATING_FROMDT_LATER_ERR

Table 9 Error Codes

Error Code	Error Code Name
104303	RCS_RATING_DATE_EQUAL_ERR
104304	RCS_RATING_CALL_ERR
104305	RCS_RATING_SEARCH_ERR
104306	RCS_RATING_SEARCH_ERR1
104307	RCS_RATING_SEARCH_RATE_ERR
104308	RCS_RATING_CURRENCY_ERR
104309	RCS_RATING_UNIT_RATE_ERR
104310	RCS_NUMERIC_ERR
104311	RCS_DATESET_OBJ_ERR
104312	RCS_UNKNOWN_ERR
104313	RCS_RATING_SEARCH_INVALID_ERR
104314	RCS_RATING_TSP_ERR
104315	RCS_RATING_TSP_ERR1
104316	RCS_RATING_STL_ERR
104317	RCS_RATING_STL_ERR1
104318	RCS_RATING_VER_ERR
104319	RCS_TSP_GET_VER_ERR
104320	RCS_STATE_ERR
104321	RCS_LOCKOUT_ERR
104322	RCS_SUSPEND_ERR
104323	RCS_DISCONN_ERR
104324	RCS_POSTACTIVE_ERR
104325	RCS_POSTACTIVE_RECHARGE_ERR
104326	RCS_INCONSIS_FAILED_CHG_CNT
104327	RCS_UNMATCH_INST_CHG_SIZE
104328	RCS_BAD_INPUT_DATA
104329	RCS_BAD_SCHEDULE
104330	RCS_NO_TERM_DEF
104331	RCS_NO_ACCOUNT_CATEGORY_REF
104332	RCS_NO_ISO_CURRENCY
104333	RCS_NO_RATE_CURRENCY_REF
104334	RCS_TAX_FAIL
104335	RCS_BULKDB_ERR
104336	RCS_ACCT_LOCKED
104337	RCS_NO_ACCOUNT_SERVER_ID
104338	RCS_ACCOUNT_SUBSCRIBER_NOT_FOUND
104339	RCS_DISCONNECT_TERMINATED_INST
104343	RCS_FETCH_RINGBUFFER_FATAL
104344	RCS_RC_AWARD_ERROR
104345	RCS_RC_RATE_OVERRIDE_ERROR
104346	RCS_SUBSCRIBER_STATUS_ERROR
104347	RCS_CMF_EXEMPT_ERROR

Table 9 Error Codes

Error Code	Error Code Name
104348	RCS_COMMITMENT_ERR
104349	RCS_INCONSIST_BAL_CHG_FATAL
104350	RCS_UNMATCHED_BAL_AMOUNT_FATAL
104351	RCS_PREPARE_DBDATA_FATAL
104352	RCS_SYSTEM_PARAM_FATAL

Maintenance Guidelines

File System Maintenance

Log Files

The Log files are placed in /staging/billing/data/log folder. The naming conventions followed for the Log files are:

ure_onDemand<server_id>.log_<timestamp>

ure_batch_<process_name>log_timestamp

Log files need to be periodically deleted to conserve disk space.

To reduce logs you can set the following masks on the `start_rcs` command line:

-trace-mask 0x08a09000 - To reduce logs (to minimum)

-fullmask - For all logs

<no input> - To keep default logging level



See http://slfwiki/wiki/index.php/Setting_desired_mask_bits for information on logging mask level changes.

Reading RCS Logs

RCS processes in multiple threads and stages. The `grep` command is used on individual threads to check if the RCS process is running correctly and to understand the RCS flow.

The stages are:

- **Preprocessing Stage** - A single thread fetches charges to be processed from the database.
[10:46:36.496-0258] RCS Preprocessing Module called
- **Segmenting Stage** - The Preprocessing thread queues messages to the segmenting thread which distributes the messages to worker or processing threads.
[10:46:36.498-1029] rcs_segmenting_thread interface initialized
[10:46:36.498-1029] rcs_segmenting_thread blocking for work
- **Processing Stage** - There are multiple threads which process charges and update database with results.
[10:46:36.498-1286] RCS processing thread [0] start
[10:46:36.499-1286] rcs_processing_thread: RCS thread [0] interface initialized
[10:46:36.499-1543] RCS processing thread [1] start

[10:46:36.500-1543] rcs_processing_thread: RCS thread [1] interface initialized
[10:46:36.500-1800] RCS processing thread [2] start
[10:46:36.501-1800] rcs_processing_thread: RCS thread [2] interface initialized

Reports

Reports are always enabled. RCS Batch Report is at the end of the RCS Batch run logs in the log file. Measurement file is available for on-demand mode. It is always created in \$ARBORCTRLRPT.

Archiving Maintenance

ARCH is ported to the rating environment and is used to archive and purge RCs and NRCs (created by RCS) periodically as these tables grow dynamically over time.

Since RCs can be generated in advance and later disconnected before their billed through dates, ensure that the RCs do not get purged too early to give CSRs the opportunity to disconnect them.

The RC and NRC tables are partitioned in the Rating DB and the RC frequency is used as a top-level partition key for the RC table.

The RCs have different frequencies. Each frequency is treated separately. Each RC frequency has its own archive type. The frequency value is populated in the ARCH_TYPE_REF column. Only one archive type is sufficient to handle NRCs since NRCs do not have different frequencies.

New archive types are added which is only used by ARCH when it is connected to a Rating database. These new archive types are present in all databases which has the ARCHIVE_TYPE_REF table and this happens due to propagation.

Other Maintenance

OnDemand RCS TCP/IP connection needs to be health checked from SAPI. If required, OnDemand RCS has to be restarted and reconnected from SAPI.

Chapter 11

Recharge Agent Process

11

orded (sender
The desti
notifying
ng The noti
ieve The m
cT access To

e

v

Recharge Agent Process Description

Recharge Agent (RA) is a process that provides uniform interface for voucher and non-voucher recharge capability through various interfaces. This includes IVRU, USSD, and SAPI.

Prerequisites for Running Recharge Agent Process

Before running the process, ensure \$ORACLE_HOME/network/admin/tnsnames.ora contains entry used by RA. By default this entry is named recharge_connection.

Running Recharge Agent Process

For IBM, Recharge Agent is started by the Unified Platform Process Management. As root issue, the following commands should be used to stop the UP agent, load the Recharge Agent config file, and start the UP agent:

```
agent stop
cp /oracle/oracle8/rchg_agent/sdsagent.cfg $JBOSS_HOME/process
agent start
```

After the UP agent comes up completely (2 minutes or so) the Recharge Agent should run and restart if it exits for any reason.

You can also do the stop and start the rechargeAgent with these two scripts:

```
sdsagentStop.sh
sdsagentStart.sh
From /oracle/oracle8/rchg_agent/
Stop the agent by running ./sdsagentStop.sh
For starting the agent ./sdsagentStart.sh
```



NOTE

Previous releases of the Recharge Agent on IBM were started via /etc/inittab. Any entries in this file should be removed.

To run sdsagent from UPA, use the following command:

```
execute_process -g database -pr sdsagent -t NI -d start
```

Configuration of Recharge Agent Process

Modify dbParameters.dat file.

Log in as oracle to active SDP node.

The configuration file dbParameters.dat can be found at /oracle/oracle8/rchg_conf.

Edit dbParameters.dat file to set the following parameters (additional parameters may be present, but need not be updated at this time):

The 1st line contains the user name to connect to the Rating database.

The 2nd line should contain the entry **recharge_connection**.

The 3rd line onwards would be some values given by <value name> = <value>.

Example:

```
sdp_rech_timeout=5
```

```
sdp_rech_agent_port=18001
```

```
sdp_rech_agent_telnet_port=18002
```

The password to connect to the rating database has to be fetched from the security server.

Provision the credentials in the security server using dbtype = RATING & instance = recharge_connection.

To fetch the password from the security server, ensure there is no line in this file starting with password=

To bypass the password fetching from the security server, you can edit the dbParameters.dat file and add an entry that is similar to this (assuming password is converse in this example)

```
password=comverse
```

To change the port the recharge agent connects to, the value has to be changed in sdp_rech_agent_port.

For example, if you want recharge agent to start as a server on port 18009, modify the entry to sdp_rech_agent_port=18009.

Be sure to change in ureCfg.xml for URE and recharge_configuration table for SAPI if you want to change this.

System Parameters

The configuration SingleAPI to RechargeAgent consists of populating the RECHARGE_CONFIGURATION database table.

The table exists within the ONLINEPC database. In a Converged deployment, this is typically the cust1 database. In a RTB Standalone deployment, this is the main database.

There must be one record in the table for every RechargeAgent configured in the system. Each record contains the hostname/port of a RechargeAgent.

Table 10 Table Structure

Column Name	Data Type	PrimaryKey Column	Req. (Y/N)	Default Value	Column Action/New Column Description
server_id	Tinyint	Y	Y	N	Primary key. Foreign key to SERVER_DEFINITION.server_id
primary_hostname	Varchar(32)	N	Y	N	Name of primary host.
primary_port	Int	N	Y	N	Primary TCP port number. e.g., 18001
backup_hostname	Varchar(32)	N	N	N	Name of backup host. For HA configurations.
backup_port	Int	N	N	N	Backup TCP port number. For HA configurations.

Optimal Recharge Agent Process Schedule

The process is started by UPA. The process can also be started on command line. The process should always be running.

Reviewing Recharge Agent Process Activity

The logs can be viewed by:

```
telnet 0 <debug port>
```

Where <debug port> is the parameter in dbPramaters.conf file

Expected Input for Recharge Agent Process

Process uses TCP/IP based XDR encoded interface, the interface specification are as defined in RechProt.x and is part of source code. All applications communicating to the process adhere to these interface specifications.

Expected Output and Location of Output

Processes the recharge request and generate response messages as per the interface specification.



See *Interface Specifications* for error codes and database error codes returned by the interface.

The table below lists the error Database Error codes returned in response messages to the client application by RA.

Table 11 Error Code and Error Description

Error Code	Error Description
0	Successful Recharge/Replenish
1	Subscriber not found
2	Offer not found
3	Offer balance details not found
4	Reseller not found
5	Subscriber in Fraud State
6	Invalid Face Value, voucher value < 0
7	Invalid Account Expiry Offset, value < 0
8	Session exceeding Recharge Limit
9	Core Balance exceeding Max Limit (only for Voucher Recharge)
10	Failed to generate recharge history details
11	Failed in public_ins_rech_hist.insert_recharge_history
12	Card doesn't exist
13	Unable to retrieve Voucher details

Table 11 Error Code and Error Description

Error Code	Error Description
14	Card in Invalid State
15	Invalid Expiration Date
16	Non service fee with Face Value 0 or different currency_unit
17	ISO mismatch between Voucher and Recharge Entity not allowed
18	Reseller conflict between Subscriber/Account and Voucher
19	Subscriber not activated due to insufficient balance
20	not activated due to insufficient balance or expired
21	Recharge History Main is Full
23	Recharge succeeded. Automatic Offer Change/Alco addition Failed during recharge
26	Failed during RCT check
27	Error while updating the recharge card when it is used
28	Unable to retrieve currency details
29	Invalid currency code
30	Card already used
31	Insert on Event History Failed
32	Invalid Voucher Currency
33	Recharge using different currency voucher not allowed
34	Invalid value in System Parameter
35	Reject recharge if recharge not allowed
36	NO ISO code defined for destination offer
37	Currency mismatch between old and new offer
39	Subscriber/Account state is not active and voucher face value is 0
50	Shadow subscriber cannot perform account recharge
51	Account in Inactive State
52	No applicable RCT found
54	Account_subscriber Record is Locked
55	Subscriber/Account doesn't have the Input Balance ID
58	Invalid Card Number Hash input
59	Invalid balance Unit Id Is Null
60	Request rejected since applicable RCT not found
61	Input subscriber is not eligible to perform Account recharge
62	Rejecting request since no balance is Affected
63	Invalid Card Code Input
65	Balance Propagation failed
66	Failed during offer swap or addition
67	Failed during the Bundle Id check for the offer swap
68	Alco offer not there in offer_ref table
69	Invalid recharge originator ID
70	AlCO end date is past to current date
41	CHECK_BAL_UNIT_TYPE Mismatch: Balance unit type not matching
71	Invalid Balance id for replenish request
72	Balance unit type id cannot be NULL for DBR

Table 11 Error Code and Error Description

Error Code	Error Description
73	Balance ID to be replenished is missing or not a pre paid balance
74	Subscriber in disconnected state cannot perform recharge
75	Invalid Face value
76	Invalid face offset
77	CHECK_CORE_REPLENISH:Core balance cannot be replenished

List of Success/Failure Criteria

From operational standpoint the interface can be determined operational at various level with level one being the least operational state to level four which is most or all functionality provided by the interface fully functioning.

1. Process is up and running can be verified using the command below -

```
sdpl:/oracle/oracle8/rchg_agent> ps -ef | grep sds
oracle8 377036      1    0 11:37:06 pts/5    0:00
/oracle/oracle8/rchg_agent
stl/sdsagent
oracle8 1286298 2502840    0 11:37:15 pts/5    0:00 grep sds
```

2. Process has successful connection to database and listening for recharge request on network interface at specified port in dbParameters.conf

```
sdpl:/oracle/oracle8/rchg_agent> telnet 0 18002
Trying...
Connected to 0.
Escape character is '^]'.
RechargeAgent> status
RechargeAgent Status
Queue Size (queued/max): 0/100
Incoming TCP Connections: 0
Oracle connect parameters: cbs_owner/comverse@recharge_connection
Threads/Oracle Connections: 4
Thread: 0, state: WAITING_FOR_REQUEST
Thread: 1, state: WAITING_FOR_REQUEST
Thread: 2, state: WAITING_FOR_REQUEST
Thread: 3, state: WAITING_FOR_REQUEST
Statistics over last 17 minutes
Current Time: [11:37:27.82]
0 requests processed at an average 0 ms each
Successful Requests: 0
Oracle Errors: 0
Data Errors: 0
Queue Overflows: 0
Highest Request Time1: 0 ms
Lowest Request Time1: 0 ms
Last Request Time1: 0 ms
```

RechargeAgent>

3. Successfully process incoming request and provide response to the requesting application

This can be verified by looking at logs at higher trace level and verifying the logs -

```
sdpl:/oracle/oracle8/rchg_agent> telnet 0 18002
```

```
RechargeAgent> mon DBG_ALL
```

```
RechargeAgent> tr
```

```
[07:59:32.24] <CN01,003>: Receive recharge request in GetRechargeHandler
```

```
[07:59:32.24] <CN01,003>: Processing Recharge from SLU 2 - SubscriberNo: 270,  
SubscriberNoResets: 0, CardNo: 837887630168
```

```
[07:59:32.27] <CN01,003>: SUCCESSFUL_RECHARGE, SubscriberNo: 270,  
SubscriberNoResets: 0, cardNo: 837887630168
```

```
[07:59:32.27] <CN01,003>: AlcsStatus: -1,CosStatus: -1
```

```
[07:59:32.27] <CN01,003>: ChangeCosF: -1,ChangeCos: 0
```

```
[07:59:32.27] <CN01,003>: FaceValue : 101.000000,ChangePhonebookF: 0
```

```
[07:59:32.27] <CN01,003>: Performing Post-Recharge Query - SubscriberNo: 270,  
SubscriberNoResets: 0
```

```
[07:59:32.27] <CN01,003>: SUCCESS Post-Recharge Query - SubscriberNo: 270,  
SubscriberNoResets: 0
```

```
[07:59:32.27] <CN01,003>: Sending Recharge response
```

```
[07:59:32.27] <CN01,003>: SERVER: XdrLen: 2116
```

```
[07:59:32.27] <CN01,003>: SERVER: Type: 2
```

```
[07:59:32.27] <CN01,003>: SERVER: Length: 2116
```

```
[07:59:32.27] <CN01,003>: SERVER: SeqNumber: 4
```

```
[07:59:32.27] <CN01,003>: SERVER: CrNumber : 98306
```

```
[07:59:32.27] <CN01,003>: SERVER: SluNumber: 2
```

```
[07:59:32.27] <CN01,003>: SERVER: TimeToLive: 0
```

```
[07:59:32.27] <CN01,003>: AJ: IEncoded: 2140
```

4. Successful subscriber account recharge reflected in Database with subscriber account recharged.

Log into the database and verify LAST_RECHARGE_DATE for the subscriber to confirm the subscriber record reflects the recharge -

```
select * from account_subscriber t where t.subscr_no=270
```

Troubleshooting Guidelines

As provided in last section for successful operation of the process verify through each step. Check dbParameters.dat file to verify Database connection details and the network port information to confirm as per the environment setup. In case of S-API verify the port information in dbParameters.dat is matching the database configuration in recharge_configuration table.

```
select * from recharge_configuration t
```

```
1      9      10.230.18.12518001
```

```
2     10     10.230.18.12518001
```

```
sdpl:/oracle/oracle8/rchg_conf> cat dbParameters.dat
```

```
cbs_owner
```

```
recharge_connection
```

```
sdp_rech_timeout=5  
sdp_rech_agent_port=18001  
sdp_rech_telnet_port=18002  
password=comverse  
sdp1:/oracle/oracle8/rchg_agent>
```


Chapter 12

Reports GUI Process

12

orded (sender
The desti
notifying
ng The noti
ieve The m
cT access To

e

Reports GUI Process Description

The main goal of the Operational Reports application is to provide backwards compatibility for the operational reports contained with the SAW for RTB 5.x. The reports are categorized into five groups based on the functionality of the reports:

1. Subscriber summary
2. Subscriber detail reports
3. Voucher reports
4. MTR reports
5. Event History reports
6. Call history reports

These reports are available in Prepaid only mode. Only Voucher reports are available in Converged mode. These reports are not available in FX only mode.

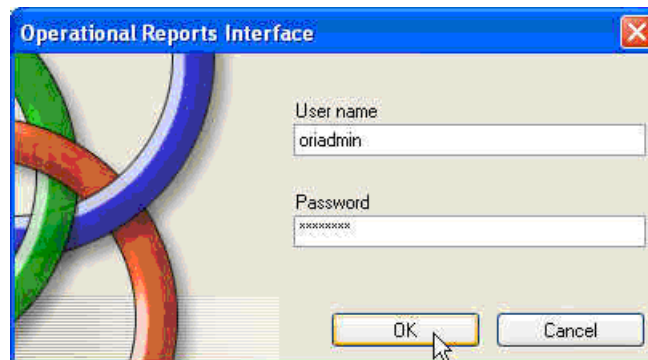


See CSR Reports chapter in Billing Reports and File Layouts User Guide for information on Converse One Reports used in Converged mode.

Login

The Main login window, handles interaction with the security server.

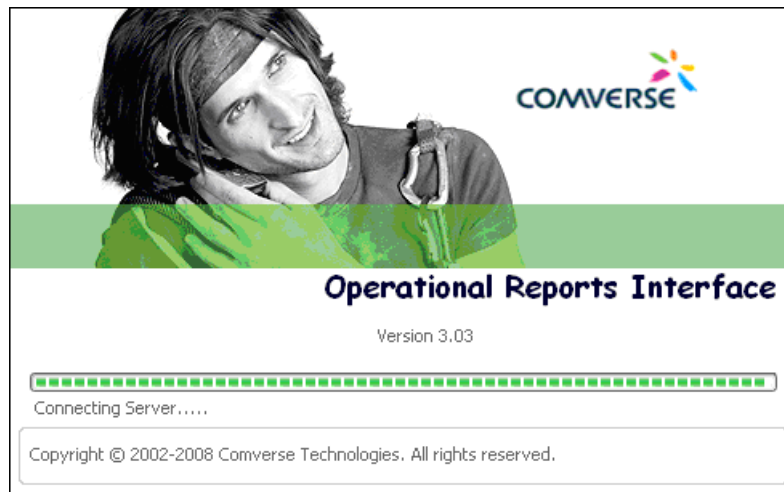
Figure 27 Operational Reports Interface – Login Window



Operational Reports Interface (ORI) Console

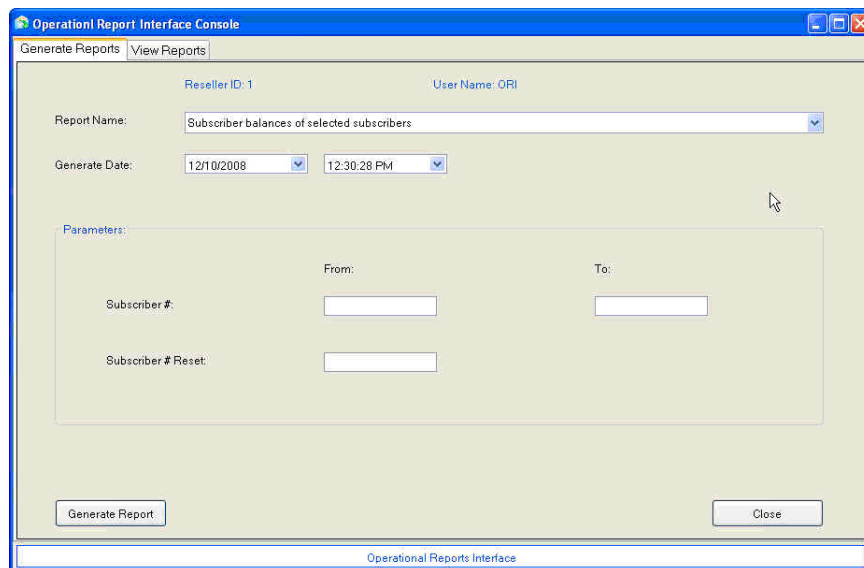
After successful authentication, the Operational Reports Interface Console window is displayed. Here the user can make a selection of the operation user wants to perform.

1. Generate New Reports – User can choose a specific report and generate it.
2. View Generated Reports – User can view a list of all generated reports and choose to either view or purge data.
3. Schedule Reports – User can choose an existing report and schedule it for a later date.

Figure 28 Operations Reports Interface

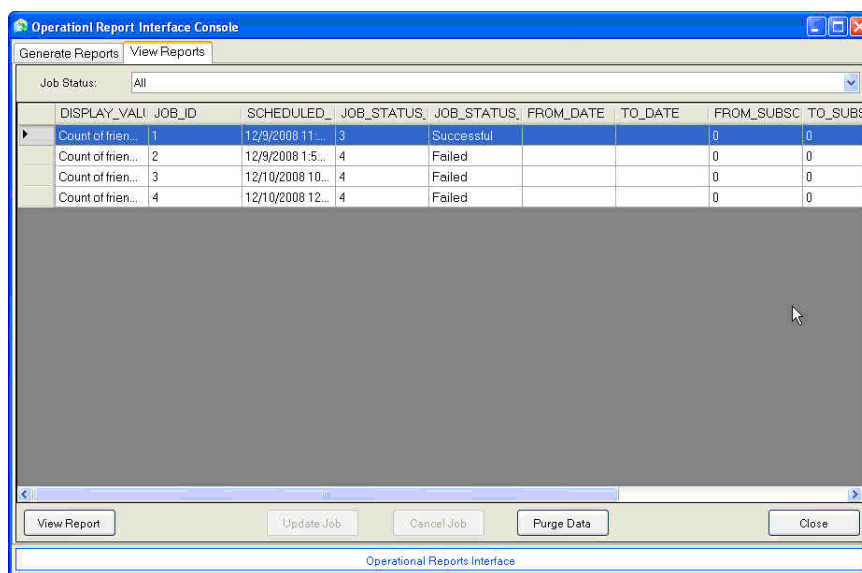
Schedule/Generate New Reports

The following screen allows the user to choose a specific report and generate it.

Figure 29 Schedule/Generate New Reports Window

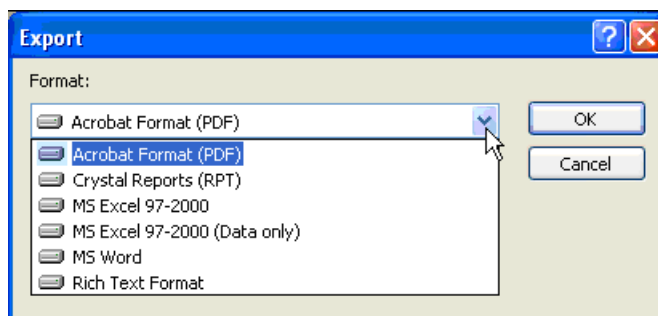
View Generated Reports

The below screen is used to view generated/processed reports.

Figure 30 View Reports Window

The user can sort on any field and view all the generated reports.

The report generated can be stored in any of the following format.

Figure 31 Report Formats

Installation

The ORI application must be installed on client machines. The installation procedure is detailed in the following sections:

Pre-required software

.NET framework 3.0 or above

Oracle Client 10.0 or above

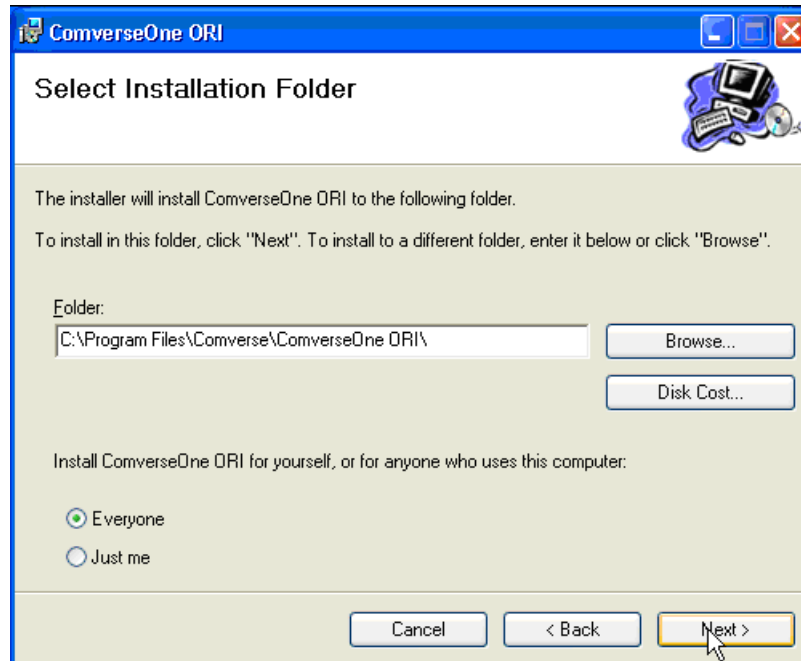
Install ORI Application

The following is the procedure for installing ORI:

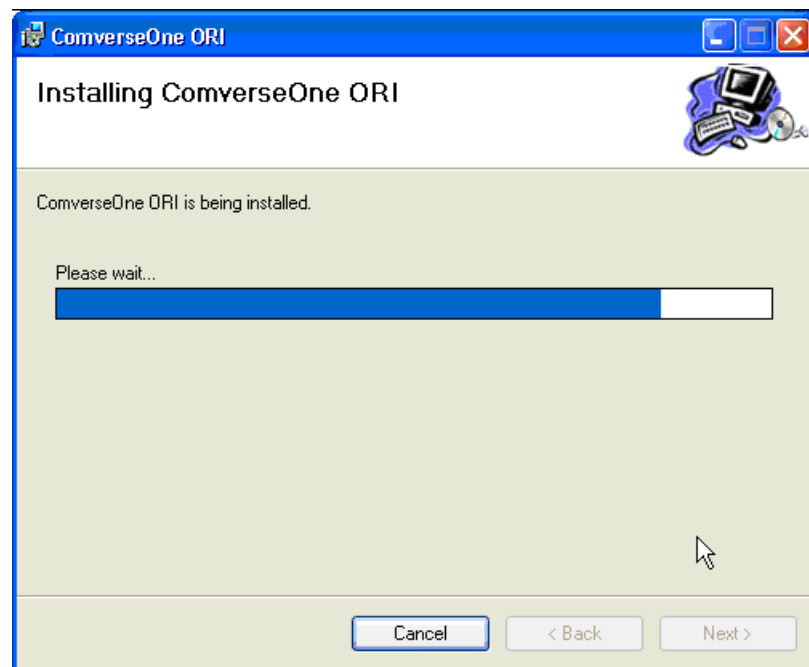
1. Execute ComverseOne.ORI.msi file and click the Next button.

Figure 32 Welcome Window

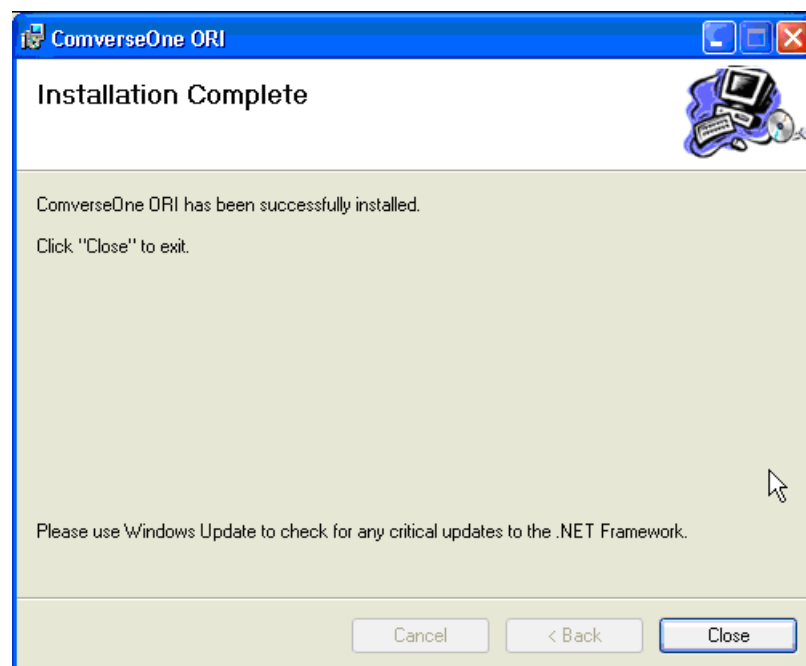
2. Choose INSTALL_DIR using **Browser** button and click the **Next** button. The default INSTALL_DIR is C:\Program Files\Comverse\ComverseOne ORI

Figure 33 Browse Install Directory

3. Click the **Next** button again.

Figure 34 Installing Comverse ONE ORI

4. Click **Close**.

Figure 35 Installation Complete

Configuration of Utility

After installing the ORI application, two configurations are required:

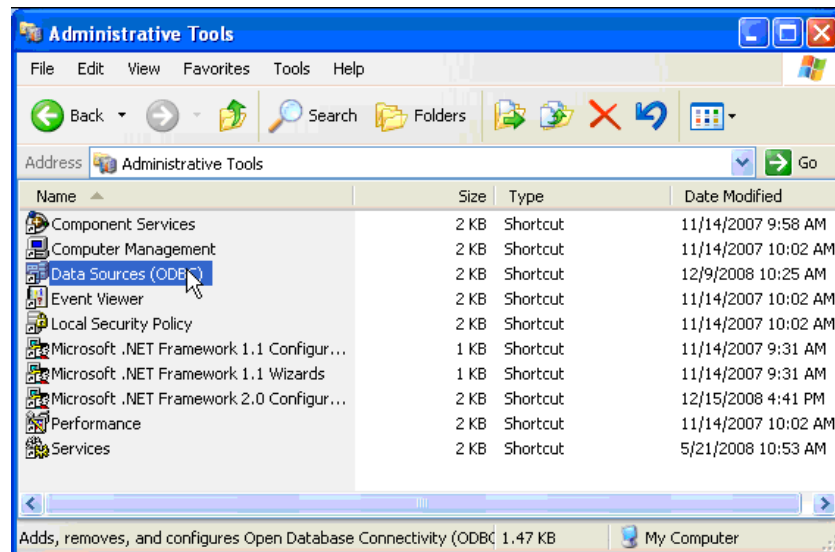
1. ODBC Configuration
2. Security server configuration for User name and password

Configuration of ODBC

This section explains the procedure for setting up ODBC for reports.

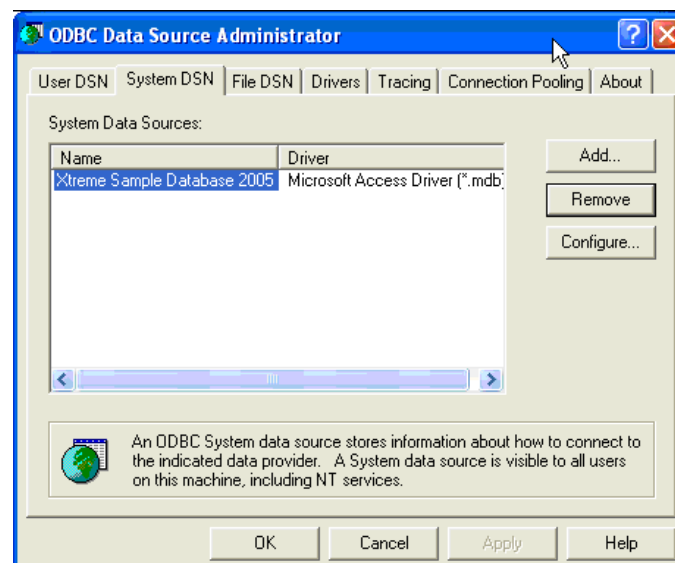
1. Open Data Sources (ODBC) from Administrative Tools in Control Panel.

Figure 36 Open Administrative Tool



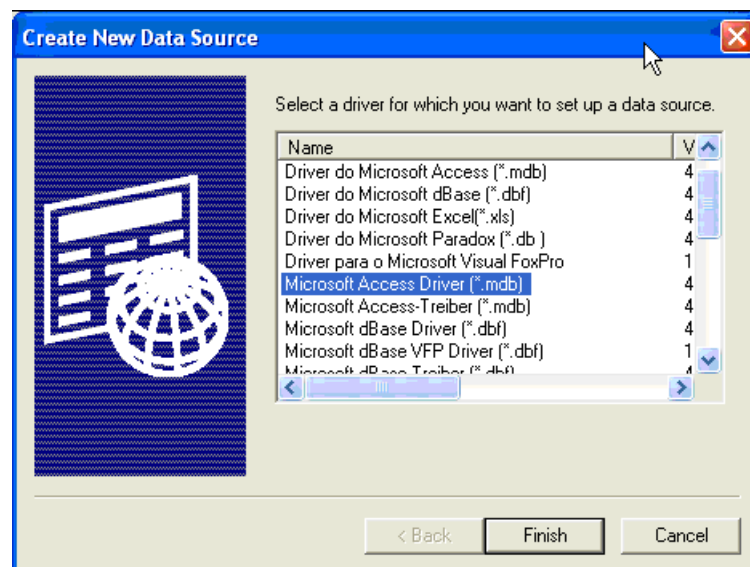
2. Go to System DSN tab and click the **Add** button.

Figure 37 System DNS Tab



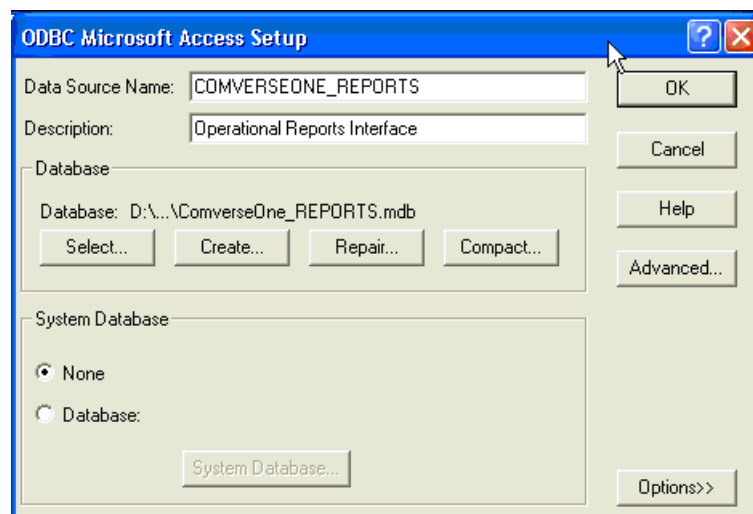
3. Select Microsoft Access Driver (*.mdb) and click the **Finish** button.

Figure 38 Create New Data Source

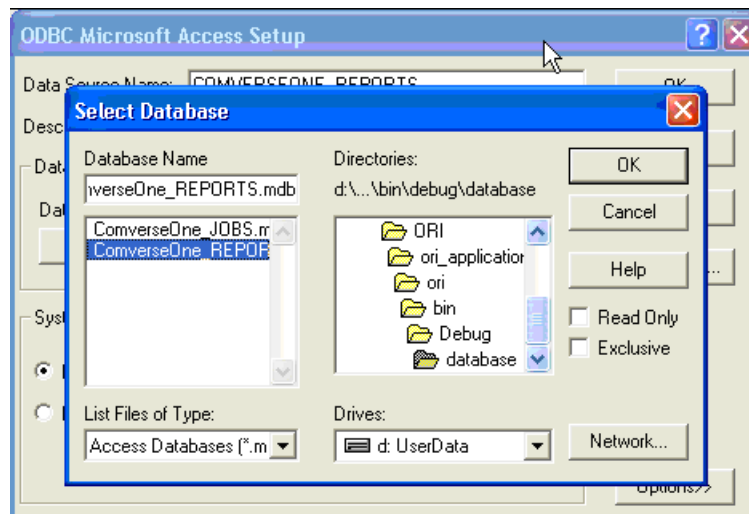


4. Input two parameters:
 - Data Source Name: COMVERSEONE_REPORTS
 - Description: Operational Reports Interface

Figure 39 Microsoft Access Driver

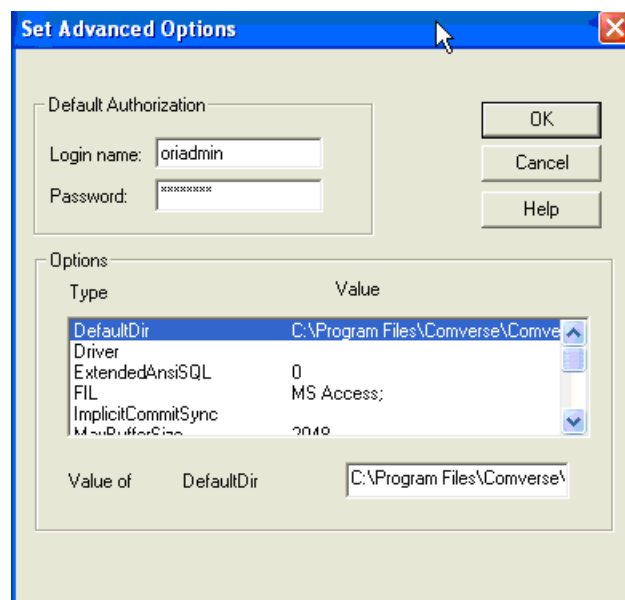


5. Click the **Select** button and choose the `INSTALL_DIR\database\ComverseOne_REPORTS.mdb` file and then click **OK**.

Figure 40 Select Database

6. Click the **Advance** button to set Default Authorization:

- ☐ Login name: **oriadmin**
- ☐ Password: **pa\$\$word**

Figure 41 Set Advanced Options

7. Click **OK** to finish the setting.

Configuring database user name and password

Database user name is configured in ORI application configuration file.

Database password is stored on security server in an encrypted format.

Configuring database user name

Database user name is stored in the ORI application configuration file `ComverseOne.ORI.exe.config` as an attribute which is `DBUserName`.



NOTE

All databases use the same user name, so there is only one attribute in the ORI application configuration file for database user name.

Configuring Database Password

Database password is stored on security server in encrypted format. It is created by using `mshell` command. If they are not created, please contact security team to create them. The following is a sample:

```
<mshell> store_credential -type database -uid cbs_owner -dbtype CBS -in  
MAIN1 -pwd converse
```



NOTE

In above sample:

Text **cbs_owner** is database user name which must match the attribute (`DBPassword`) value in the ORI application configuration file.

Text **CBS** must match the attribute (`DBType`) value in ORI application configuration file.

Text **MAIN1** is the database SID or service name. This value comes from `DSQUERY` field in server definition table.

Text **converse** is the database password. Different database can have different password.

At least database type 9, 10 and 11 passwords has to be created in security server.

Configuration of SAPI server IP and Security Server IP

Update SAPI server IP and Security Server IP addresses in file `ComverseOne.ORI.exe.config`.

Configuration of Voucher Database

By default the Voucher Database is configured to be on the rating DB. Thus the voucher reports #9 and #10 is pulled from server ID 9 (Rating DB).

In deployments where the CVS is on a different DB, the User needs to specify the separate voucher server ID in the properties file after installation.

Configure CVS Server IDs

If the Voucher server is standalone in the deployment, then the user needs to manually update CVS server id in the file `ComverseOne.ORI.exe.config`.



NOTE

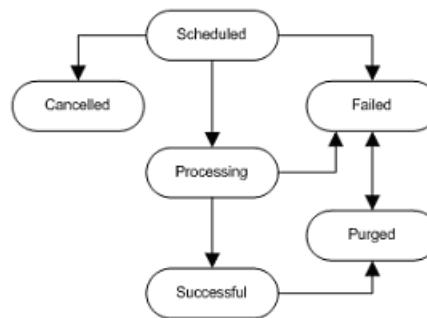
A comma sign should be used if there are more than one CVS. For instance, 9, 10.

Environmental Variables

The following parameters are configurable:

1. DB Information
2. Cron Job Schedule Time
3. Log File

Figure 42 Report Status Data Loader Process State Transitions



The above figure shows the Report Status Data Loader Process State Transitions. A report job can take the above states. When the report job is successful, the report is automatically loaded on to the user terminal for display.

System Parameters

Deployment Mode Support

The system picks up the SYSTEM_PARAMETER (PARAMETER_NAME = 'DEPLOYMENT_MODE') and based on its value it behaves as follows:

- DEPLOYED_MODE_UNDEFINED = 0;
 - The Reports Drop down list shows **no** reports
- DEPLOYED_MODE_PREPAID = 1;
 - The Reports Drop down list shows **ALL** reports
- DEPLOYED_MODE_PREPAID_ENHANCED = 2;
 - The Reports Drop down list shows **ALL** reports
- DEPLOYED_MODE_CONVERGED = 3;
 - The Reports Drop down list shows report **#9 and #10**
- DEPLOYED_MODE_CONVERGED_ENHANCED = 4;
 - The Reports Drop down list shows report **#9 and #10**
- DEPLOYED_MODE_POSTPAID = 5
 - The Reports Drop down list shows **no** reports
- DEPLOYED_MODE_POSTPAID_ENHANCED = 6;
 - The Reports Drop down list shows **no** reports

Optimal Process Schedule

To avoid performance issues, it is highly recommended to schedule reports during off peak hours. While running on demand reports, care should be taken to ensure that the reports parameters and date ranges are set to a reasonable value to avoid performance issues.

Reviewing Process Activity

The log files are located in the installation directly.

Expected Input for Process

The User can select and choose from a set of predefined 13 reports.

Table 12 Reports

Report #	Report Name	Input Parameters	Database
RPT_001	Summary of subscribers by current state and primary offer	No Parameter. (All records)	ALL SDP
RPT_002	Summary of subscribers by primary offer and current state	No Parameter. (All records)	ALL SDP
RPT_004	Count of friends and family subscribers grouped by primary offer.	No Parameter. (All records)	ALL SDP
RPT_005	Subscribers with current status of "53"	No Parameter. (All records)	ALL SDP
RPT_006	List of activated subscribers	Active Date Range	ALL SDP
RPT_007	Subscriber balances of selected subscribers	Subscriber Number Range	ALL SDP
RPT_008	Subscriber list of selected subscriber IDs	Subscriber Number Range	ALL SDP
RPT_009	Count of unused vouchers for a face value range(No Changes)	Face value Range	MASTER_SDP
RPT_010	Count of all consumed vouchers (No Changes)	Used Date Range	MASTER_SDP
RPT_011	Call History for selected subscribers	Subscriber Number Range	MASTER_SDP or HISTORY DB
RPT_012	Monetary transaction audit report	Date range	MASTER_SDP or HISTORY DB
RPT_013	Post active subscriber remaining balance removal report for selected dates	ModDate Range	MASTER_SDP or HISTORY DB
RPT_014	Event History – potential recharge fraud within a selected date range	Date range	MASTER_SDP or HISTORY DB



The application must be running on the user machine if the user wants to run scheduled jobs. The scheduler is internal to the application and is invoked only when the application is running.


Expected Output and Location of Output

An example output of the report is shown below.

Figure 43 Report Output

List Of Activated Subscribers For A Selected Date Range Sorted By Subscriber Number	
Subscriber Number	Date Activated
1	
3	8/27/2008
4	
5	9/1/2008
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	8/28/2008
24	
25	
26	
27	
28	
29	
30	
31	
32	
33	
34	
35	

Run Date: 14 November 2008 Page 1 Prepaid Service: 11-008



Troubleshooting Guidelines

FAQs

1. Cannot log in to the application?
Check for the following:
 - ☐ SAPI 3 tier server configuration
 - ☐ Security server configuration
2. Scheduled report is not being picked up for processing?
Check for the following:
 - ☐ Check DB configuration properties
 - ☐ Check for the scheduled time

Maintenance Guidelines

The Database and Archiving maintenance guidelines are given below:

Database Maintenance

Note that the Application uses MS Access DB to store data for the reports. This DB needs to be maintained for performance and has size limitations.

Archiving Maintenance

The purge functionality ensures that data is deleted from the MS access DB and the file size is also reduced. However it is recommended that the reports are generated with a limited date range. The reports can be saved in various formats based on the preference of the user and limited by the file system space.

Chapter 13

UFA Process

13

orded (sender
The desti
notifying
ng The noti
ieve The m
t access To

e

UFA Process Description

UFA is a utility module for the Database maintenance. When this module runs, it archives/purges the designate usage tables in the database. The following tables are archived/purged by UFA:

- FILE_STATUS
- USAGE_FILE_COUNTS
- USAGE_FILE_COUNTS_HISTORY
- USAGE_LOCAL_COUNTS
- USAGE_LOCAL_ROLLUP
- USAGE_FILE_STATUS
- USAGE_FILE_STATUS_HISTORY

Prerequisites for Running UFA Process

There is no prerequisite for running UFA. Run the UFA from command line.

Dependencies on Other Processes

No dependencies.

Running UFA Process

UFA is designed to run from the command line. Full command as below:

UFA <process_name> <unscaled server ID>

Where,

- **Process_name** - process name to be run: **ufa01**.
- **Unscaled server ID** - server ID of BLUS (on Converge = 4) or ORP (on Realtime = 12) DB.

Configuration of UFA Process

There is no configuration file.

Environmental Variables

The Environmental Variables are:

- ARBORDIR - Top level directory where the applications are installed.
- ARBORLOG - Path to where the log file should go to (i.e. /staging/billing/log)
- ARBORCTRLRPT - Path to where the control report file should go to (i.e. /staging/billing/reports)
- ARBOR_CATALOG_DATABASE - Catalog database server (CTLG for Converge, MAIN for Real time mode)
- ARBOR_CATALOG_QUERY - Catalog database instance (CTLG for Converge, MAIN for Real time mode)
- ARBORDATA - Top level directory where the usage files located (i.e. /staging/billing/data)
- ARBORBIN - Where all the binaries are located.
- ARBORDBU - User schema to log in database (i.e cbs_owner)

- OAM_ENV_CONN_MA:
 - = TRUE (to connect to security server to get database password)
 - = FALSE (Not to use security server for database password)
- RATE_DT_DATETIME - User specifies datetime to be used with UFA, instead of getting datetime from the system. Setting value in this format: YYYYMMDDHHMISS

System Parameters

The system parameters are:

- UFA/ARCHIVE_DAYS: All rows which have ARCHIVE_DAYS older than the current comparison date are archived. Default value is 30 days (To be used on BLUS)
- UFA/PURGE_AFTER_N_DAYS: All rows which have PURGE_AFTER_N_DAYS older than the current comparison date are purged. Default value is 60 days (To be used on BLUS)
- UFA/OR_ARCHIVE_DAYS: All rows which have OR_ARCHIVE_DAYS older than the current comparison date are archived. Default value is 30 days (To be used on ORP)
- UFA/OR_PURGE_AFTER_N_DAYS: All rows which have OR_PURGE_AFTER_N_DAYS older than the current comparison date are purged. Default value is 30 days (To be used on ORP)
- UFA/USE_INLINE: Use SQL inline or not. Value is 1 or 0. Default = 1

Optimal UFA Process Schedule

This process is run on demand.

Reviewing UFA Process Activity

The log file is located at \$ARBORDIR directory with the format:

```
<process_name>-<date>-<time>-<server_id>-<random_number>.BATCH.000
```

List of Success/Failure Criteria for UFA Process

After a successful run, UFA returns to the prompt normally. Any error message that prints out to the screen is considered as a failure.

Troubleshooting Guidelines

Error

Cannot log into database

Solution

1. If environment OAM_ENV_CONN_MA=TRUE, then check the security server connectivity.
2. If environment OAM_ENV_CONN_MA=FALSE, then check if the encrypt password file exist or not.
3. If a) and b) is fine, then check the environments below to see if they are correctly set:
 - ARBOR_CATALOG_DATABASE
 - ARBOR_CATALOG_QUERY

ARBORDBU

4. The ARBOR_CATALOG_DATABASE and ARBOR_CATALOG_QUERY environment variables should be set as follows:

For CV mode:

ARBOR_CATALOG_DATABASE=CTLG

ARBOR_CATALOG_QUERY=CTLG

For RT mode:

ARBOR_CATALOG_DATABASE=MAIN

ARBOR_CATALOG_QUERY=MAIN

Error

Database interaction error

Solution

If table schema is different than what UFA expected, then check if UFA is running on different version of production (i.e. UFA built on Comverse ONE 3.0 to be run on Comverse ONE 3.1 environment).

In normal run, UFA log contains the information given below:

```
ngut01@drdibm17:/usr1/arbor/bin/billing/GA31_TC9/data/log 376 % cat ufa01-090416-145516-04-1392704.BATCH.000
```

```
20090416 14:55:25.280 (I 0000000) [0][0][001] EMIT_BYPASS Value of environment variable OAM_ENV_CONN_MA: 'FALSE'
```

```
20090416 14:55:25.280 (I 0000000) [0][0][001] EMIT_BYPASS Value of environment variable OAM_ENV_RECONN_ATTEMPT_NUMBER: '10'
```

```
20090416 14:55:25.280 (I 0000000) [0][0][001] EMIT_BYPASS Value of environment variable OAM_ENV_RECONN_ATTEMPT_INTERWAIT: '30'
```

```
20090416 14:55:31.171 (I 0000000) [0][0][001] EMIT_BYPASS The following system parameter(s) are either ...
```

```
20090416 14:55:31.171 (I 0000000) [0][0][001] EMIT_BYPASS Be set or using default value if not set:
```

```
20090416 14:55:31.171 (I 0000000) [0][0][001] EMIT_BYPASS System parameter UFA/ARCHIVE_DAYS = 30
```

```
20090416 14:55:31.171 (I 0000000) [0][0][001] EMIT_BYPASS System parameter UFA/PURGE_AFTER_N_DAYS = 60
```

```
20090416 14:55:31.171 (I 0000000) [0][0][001] EMIT_BYPASS System parameter UFA/OR_ARCHIVE_DAYS = 30
```

20090416 14:55:31.171 (I 0000000) [0][0][001] EMIT_BYPASS System parameter UFA/OR_PURGE_AFTER_N_DAYS = 30

20090416 14:55:31.171 (I 0000000) [0][0][001] EMIT_BYPASS System parameter UFA/USE_INLINE = 1

20090416 14:55:31.308 (I 0000000) [0][0][001] EMIT_BYPASS

===== UFA Report =====

20090416 14:55:31.308 (I 0000000) [0][0][001] EMIT_BYPASS Begin archiving *FILE_STATUS tables
.

20090416 14:55:31.706 (I 0000000) [0][0][001] EMIT_BYPASS Successfully remove 7 row(s) from USAGE_LOCAL_ROLLUP on Cust server_id 3

20090416 14:55:31.858 (I 0000000) [0][0][001] EMIT_BYPASS Successfully remove 7 row(s) from USAGE_LOCAL_COUNTS on Cust server_id 3

20090416 14:55:32.052 (I 0000000) [0][0][001] EMIT_BYPASS Successfully remove 7 row(s) from USAGE_LOCAL_ROLLUP on Cust server_id 5

20090416 14:55:32.201 (I 0000000) [0][0][001] EMIT_BYPASS Successfully remove 7 row(s) from USAGE_LOCAL_COUNTS on Cust server_id 5

20090416 14:55:32.419 (I 0000000) [0][0][001] EMIT_BYPASS Successfully remove 7 row(s) from USAGE_LOCAL_ROLLUP on Cust server_id 6

20090416 14:55:32.556 (I 0000000) [0][0][001] EMIT_BYPASS Successfully remove 7 row(s) from USAGE_LOCAL_COUNTS on Cust server_id 6

20090416 14:55:32.903 (I 0000000) [0][0][001] EMIT_BYPASS Successfully archive 7 row(s) from USAGE_FILE_STATUS on unscaled server

20090416 14:55:32.904 (I 0000000) [0][0][001] EMIT_BYPASS Successfully archive 7 row(s) from USAGE_FILE_COUNTS on unscaled server

20090416 14:55:32.904 (I 0000000) [0][0][001] EMIT_BYPASS Successfully archive 7 row(s) from FILE_STATUS on unscaled server

20090416 14:55:33.127 (I 0000000) [0][0][001] EMIT_BYPASS End archiving *FILE_STATUS tables
.....

20090416 14:55:33.127 (I 0000000) [0][0][001] EMIT_BYPASS Begin cleaning up File HISTORY tables
.....

20090416 14:55:33.341 (I 0000000) [0][0][001] EMIT_BYPASS End cleaning up File HISTORY tables
.....

20090416 14:55:33.341 (I 0000000) [0][0][001] EMIT_BYPASS

===== End UFA Report =====

ngut01@drdbm17:/usr1/arbor/bin/billing/GA31_TC9/data/log 377 % cat ufa01-090416-145540-12-594244.BATCH.000

20090416 14:55:50.518 (I 0000000) [0][0][001] EMIT_BYPASS Value of environment variable OAM_ENV_CONN_MA: 'FALSE'

20090416 14:55:50.518 (I 0000000) [0][0][001] EMIT_BYPASS Value of environment variable OAM_ENV_RECONN_ATTEMPT_NUMBER: '10'

20090416 14:55:50.518 (I 0000000) [0][0][001] EMIT_BYPASS Value of environment variable OAM_ENV_RECONN_ATTEMPT_INTERWAIT: '30'

20090416 14:55:56.456 (I 0000000) [0][0][001] EMIT_BYPASS The following system parameter(s) are either ...

20090416 14:55:56.456 (I 0000000) [0][0][001] EMIT_BYPASS Be set or using default value if not set:

20090416 14:55:56.456 (I 0000000) [0][0][001] EMIT_BYPASS System parameter UFA/ARCHIVE_DAYS = 30

20090416 14:55:56.456 (I 0000000) [0][0][001] EMIT_BYPASS System parameter UFA/PURGE_AFTER_N_DAYS = 60

20090416 14:55:56.456 (I 0000000) [0][0][001] EMIT_BYPASS System parameter UFA/OR_ARCHIVE_DAYS = 30

20090416 14:55:56.456 (I 0000000) [0][0][001] EMIT_BYPASS System parameter UFA/OR_PURGE_AFTER_N_DAYS = 30

20090416 14:55:56.456 (I 0000000) [0][0][001] EMIT_BYPASS System parameter UFA/USE_INLINE = 1

20090416 14:55:56.622 (I 0000000) [0][0][001] EMIT_BYPASS

===== UFA Report =====

20090416 14:55:56.622 (I 0000000) [0][0][001] EMIT_BYPASS Begin cleaning up *FILE_STATUS tables

20090416 14:55:57.330 (I 0000000) [0][0][001] EMIT_BYPASS End cleaning up *FILE_STATUS tables

20090416 14:55:57.330 (I 0000000) [0][0][001] EMIT_BYPASS

===== End UFA Report =====

ngut01@drdibm17:/usr1/arbor/bin/billing/GA31_TC9/data/log 378 %

Chapter 14

URE Process

14

orded (sender
The desti
notifying
ng The noti
ieve The m
cT access To

e

URE Process Description

The Unified Rating Engine process is responsible for handling the Rating requests.

Online URE works as a totally event driven process. It does not maintain a state, but instead provides the functionality by taking actions based on incoming messages. In this mode, OMNI is the messaging interface. Messages come from the SLF applications like CallProcessor (OPPS/TPPS), USSD, GPRS, Cap3Sms, Diameter, OSA, and ECI.

Offline URE works as a task driven process. It maintains state, recording the progress of its tasks in the Database. In place of a messaging interface, Offline URE loads a C-CAP module. The C-CAP module controls the process, performing its tasks as it processes Usage Records. The C-CAP processing is detailed in later sections.

Prerequisites for Running URE Process

The detailed steps for the scratch installation are given below:

1. Install OS.
2. After installation of OS is done, reboot the machine and copy the RPM's mentioned in the Installation Media/Package details section step #2 to /data/install directory. Create the directory /data/install if it does not exist.
3. Make sure 3p entities are installed which are mentioned in Install 3P step above.
4. Install RPMs. URE process is a part of installer for CBS_SLU_BASE-3.x.x. Rating_Common-3.x.x and and Rating_Tsp-3.x.x must be installed before Rating_Ure-3, x.x.
5. Run configureSLU.



See [“Configuration of URE Process”](#) section for instructions.

6. Configure security server.
 - a. Install the SEC_API rpm.
 - b. Also, add security server IP to /etc/hosts, if it is not already there, with hostname **secserv** (the IP needs to be the actual IP of the security server).
 - c. Use the following parameters to provision an entry on the security server for Rating as well as Billing DB:

Username: <the User name for DB>
 Password: <the Password for DB>
 Default DB Instance: <instance_name>
 Default DB Type: <DB_Type>
 - d. The `commonCfg.xml` configuration file in /home/omni/conf directory should contain the actual user name and database instance of the Rating DB in **DefaultOracleUserName** and **DefaultDatabaseInstance** tags respectively.
 - e. The `ureCfg.xml` configuration file in /home/omni/conf directory should contain the actual usernames and instances of Calling Circle DB and Billing DB in **CcDbUserName**, **DefaultCcDbInstance**, **BILLING_DB_USER_NAME**, and **BILLING_DB_INSTANCE** tags.

**NOTE**

Billing DB information is used only on offline rater.

Dependencies on Other Processes

URE needs to register to TSP to get static data loaded.

**SEE**

See [“Configuration of URE Process”](#) section.

Running URE Process

To start OMNI, perform these actions:

1. Log in as root, using **su** password **sonora**
2. Start OMNI with the `/sbin/service omni start` command.

To stop OMNI, use this command: `/sbin/service omni stop`

Configuration of URE Process

The detailed steps for the configuration of URE are given below:

1. Modify `cestart.202` and `tnsnames.ora` as per each setups requirement. A typical `cestart.202` file is shown below:

```
OPPS . 1 DEF DEF 202 100 /home/omni/bin/CallProcessor -name OPPS -T
TPPS . 1 DEF DEF 202 100 /home/omni/bin/CallProcessor -tpps -name TPPS -T
PMTSVR . 1 DEF DEF 202 101 /home/omni/bin/PmtSvr -d -hbt 180
URE_Q1 . 1 DEF DEF 202 100 /home/omni/bin/ure -Q -ratingdb 1
URE_U1 . 2 DEF DEF 202 100 /home/omni/bin/ure -U -ratingdb 1
URE_Q9 . 1 DEF DEF 202 100 /home/omni/bin/ure -Q -ratingdb 9
URE_U9 . 2 DEF DEF 202 100 /home/omni/bin/ure -U -ratingdb 9
TSP . 1 DEF DEF 202 100 /home/omni/bin/tsp
USSD_SLF . 0 DEF DEF 202 100 /home/omni/bin/ussd_slf
SLF_SMS . 1 DEF DEF 202 100 /home/omni/bin/cap3sms
```

2. The following URE configuration files are installed as part of the SLF rpm(s) (all in `/home/omni/conf`):

```
commonCfg.xml
ureCfg.xml
tspCfg.xml
```

The oracle usernames and instances for various DB should be updated in the XML configuration files (see [“Prerequisites for Running URE Process”](#) section).

The values contained in these files as well as in other applications `<application_name>Cfg.xml` can be customized as needed.

XML Configuration Files

The application specific XML files and respective XSD files generated during build time are mentioned in the table below.

Table 13 XML files and XSD files

XML	XSD
CallProcessorCfg.xml	CallProcessorCfg.xsd
cap3smsCfg.xml	cap3smsCfg.xsd
commomCfg.xml	commomCfg.xsd
notifAgentCfg.xml	notifAgentCfg.xsd
npqCfg.xml	npqCfg.xsd
osasoapCfg.xml	osasoapCfg.xsd
pmtServerCfg.xml	pmtServerCfg.xsd
slf_gprsCfg.xml	slf_gprsCfg.xsd
tspCfg.xml	tspCfg.xsd
ureCfg.xml	ureCfg.xsd
ussd_slfCfg.xml	ussd_slfCfg.xsd

The parameters in these XML files are set to known default values. In most of the cases the default values supplied are correct. Some parameter values such as CcDbUserName, DefaultOracleUserName and so on might have to be changed during the configuration. Some system dependent parameter values such as NumThreads, NumOfConnectionToSdp and so on in ureCfg.xml and activecalls in oppsCfg.xml are computed based on system parameters such as memory and so on. The system dependent values are most optimized values for a given system and hence they are hidden from the user. If these values have to be modified then these values have to be manually inserted into their respective XML files. Modifying system generated parameter values is not recommended. Note that the values supplied in the XML file have precedence over the system computed values.

Each parameter has its own data type (integer, string, bool, dialpadchar, char, and so on) and maximum and minimum bounds for appropriate types. The supplied XSD files contain the XML schema definitions (XSD) that can be used to validate the XML files on any SLU host machine or management machines (UPM) where the XML configuration files for SLF are stored and manipulated.

XML file structure and XSD

The XML structure is as defined below:

- There is one XML (and its corresponding XSD) for each application. The format of XML structure is <application><instance><Parameter>.
- The commonCfg.xml file is common for all applications.
- Differentiation between multiple instances of the application and the application instance ID is done.

For example, URE_U and URE_Q are instances of the URE application. But URE_U7 and URE_U28, are just the same URE_U instance in different identities. While URE_Q and URE_U are instances from the same URE binary, their inner logics are different and potentially can use different configuration parameters and/or different validation criteria (the same goes for TPPS and OPPS). The multiple identities from the same instance, for example, URE_U1 and URE_U2, follow the same logic and should share the same group of configuration parameters and same validation criteria. Therefore, for multiple instances, use different XML element names. For example,

<OPPS> and <TPPS>. But for multiple identities of the same instance, use the id attributes. For example, use <URE_U id=12> for URE_U12. Reserve ID=0 to denote parameters intended for all identities.

- d. Under the root application node, there is the <ALL> instance node which contains parameters intended for all instances. There is one node for each instance of the application. The sequence of the instance nodes are enforced so as to make sure the <ALL> instance node comes first.
- e. For ure, unlimited numbers of multiple identities are allowed for both URE_Q and URE_U. But the uniqueness of the ID is enforced.
- f. Inside each instance node, the parameter elements can appear in any order. All parameters are allowed to be missed, but no duplicates are allowed.
- g. At runtime a specific application instance (and for URE, with a specific instance and rating DB id's) reads all nodes applicable to itself. For example, in example below, URE3_Q2 gets:

par2=22 (from <ure> <URE_Q insid="3" dbid="2">)

par3=30 (from <ure> <URE_Q insid="0" dbid="0"> because id="0" is for all ID's)

par4=42 (from <ure> <URE_Q insid="3" dbid="2">)

par6=6 (from <ure> <ALL>).

(The value of par5 is the default value from SlfConfig.dat, initialized in the generated code, because par5 is not defined in any applicable nodes in the XML file).

```
<ure>
  <ALL>
    <par2>2</par2>
    <par3>3</par3>
    <par6>6</par6>
  </ALL>
  <URE_Q insid="0" dbid="0">
    <par2>20</par2>
    <par3>30</par3>
  </URE_Q>
  <URE_Q insid="3" dbid="2">
    <par2>22</par2>
    <par4>42</par4>
  </URE_Q>
  <URE_Q insid="3" dbid=5>
    <par2>25</par2>
    <par5>78</par5>
  </URE_Q>
</ure>
```

System Parameters

Table 14 System Parameters

Module	Parameter Name	Default	Use
CBS	ANUM_VPNCODE_CL	0	Passes A-Number short code to IPF. <ul style="list-style-type: none">■ 0=disabled■ 1=enabled
SLF	SLF_CUG_ANUMBER_ENHANCE	1	Allows SLF composed A-Number as the concatenated access code and short code for inter-CUG call. <ul style="list-style-type: none">■ 0=disabled■ 1=enabled

Chapter 15

SAPI Process

15

orded (sender
The desti
notifying
ng The noti
ieve The m
cT access To

e
v

SAPI Process Description

SAPI is a middleware system which provides EJB/Webservice interfaces calling by all Front End clients' components to communicate with backend servers and databases, and provides services of management Account/Subscriber, Order, Offer, Bundle etc.

Dependencies on Other Processes

Dependencies are:

- Database must be installed and configured
- UPM/UPA must be installed and started before starting SAPI

Running SAPI Process

Full command line syntax with explanation of all command-line switches:

From Command Line:

Start the SAPI server.

1. Login with apiuser.
2. Go to the installation dir/server/bin.
3. Run start_sapi_admin script.
4. Run start_managed_sapi script.

Example:

```
cd /home/sapi_wl/server/bin
./start_sapi_admin.sh
./start_managed_sapi.sh
```

Stop the SAPI server.

1. Login with apiuser.
2. Go to the installation dir/server/bin.
3. Run stop_sapi_admin script.
4. Run stop_managed_sapi script.

Example:

```
cd /home/sapi_wl/server/bin
./stop_managed_sapi.sh
./stop_sapi_admin.sh
```

Environmental Variables

The Environmental variables are:

- Need login apiuser
- CBS_SAPI_SERVER_INSTALLDIR should be set up with SAPI installation dir

Example:

```
CBS_SAPI_SERVER_INSTALLDIR=/home/sapi_wl/server
```

- JAVA_HOME

- BEA_HOME and WL_HOME
- ORACLE_HOME

System Parameters

PO_SWAP_NOTIF

This system parameter controls the generation of notification whenever there is a change in the primary offer.

The values are:

- 0 - Disable generation of notification
- 1 - Enable generation of notification

The default value is 0.

Optimal SAPI Process Schedule

As required.

Reviewing SAPI Process Activity

Log is under <SAPI_domain>, for example:

```
/home/sapi_wl/server/domain/start_sapi.log  
/home/sapi_wl/server/domain/stop_sapi.log  
/home/sapi_wl/server/domain/SAPI_Logs/SingleAPI.log
```

Expected Output and Location of Output

After starting SAPI, the output is as shown below, if successful:

```
> ./start_sapi_admin.sh  
  
STATUS=0  
PID=14938  
INFO=<May 7, 2009 2:14:24 PM MDT> <Notice> <WebLogicServer> <BEA-000360> <Server  
unscaled server started in RUNNING mode>
```

After stopping SAPI, it gives the below output if successful:

```
> ./stop_sapi_admin.sh  
  
STATUS=0  
INFO=Done
```

List of Success/Failure Criteria for SAPI Process

If the process runs successfully, then the output is as described in the above section.

You also can check if SAPI process is running or not using:


```
ps -ef|grep java
```

If running start/stop script fails, then the error information is displayed. For example:

```
> ./start_sapi_admin.sh
```

Error: Current user is not apiuser

```
> ./stop_sapi_admin.sh
```

STATUS=1

*INFO=WLSTException: Error occurred while performing connect: Error getting the initial context.
There is no server running at t3://10.8.32.42:8800 Use dumpStack() to view the full stacktrace*

Troubleshooting Guidelines

The guidelines are:

1. Check if SAPI process is running or not using:

```
ps -ef|grep java
```

2. Go to SAPI domain, check logs, for example:

```
/home/sapi_wl/server/domain/start_sapi_admin.log
```

```
/home/sapi_wl/server/domain/stop_sapi_admin.log
```

```
/home/sapi_wl/server/domain/start_managed_sapi.log
```

```
/home/sapi_wl/server/domain/stop_managed_sapi.log
```

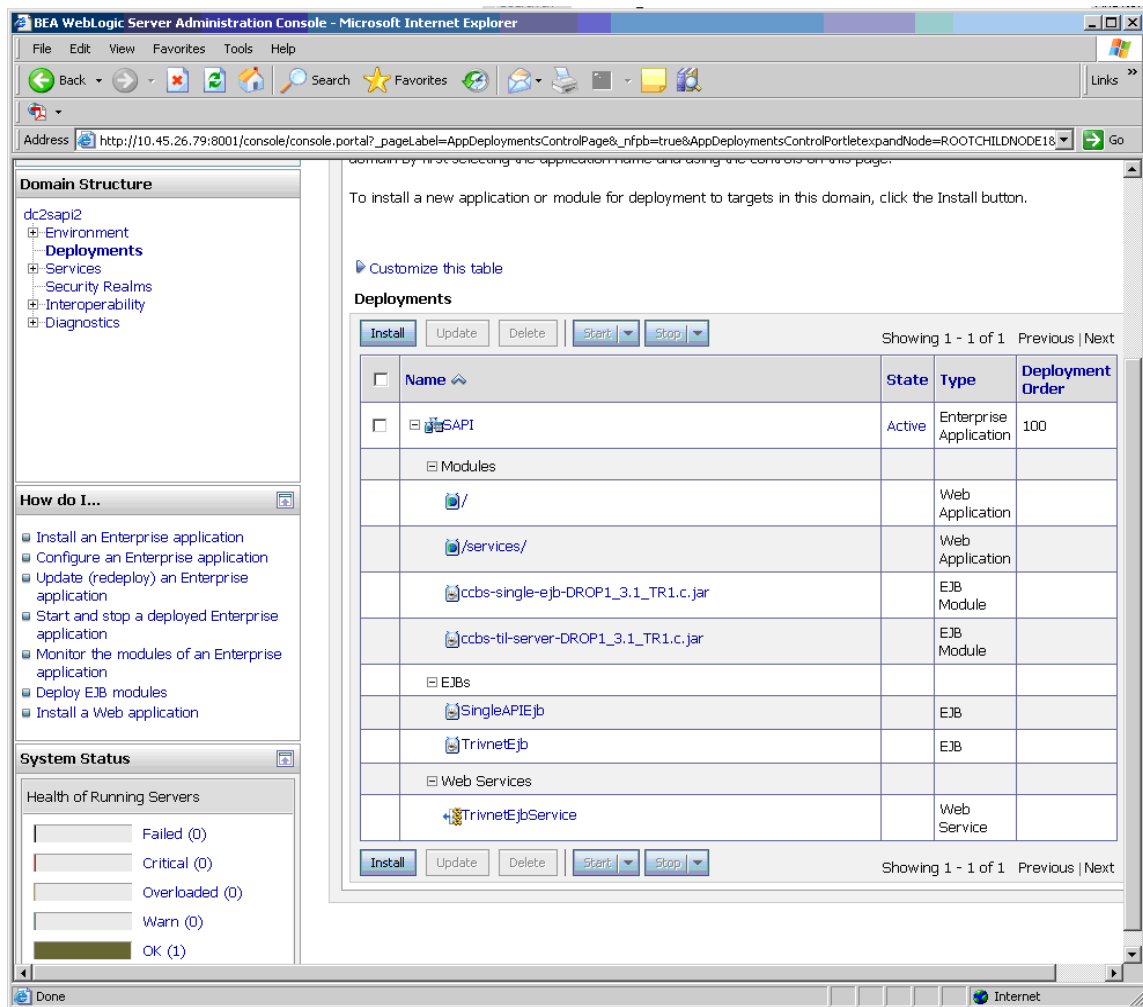
```
/home/sapi_wl/server/domain/SAPI_Logs/SingleAPI.log
```

3. Go to SAPI Admin Console:

- ❑ Open the web browser, and access http://<sapi_host>:<sapi_port>/console.
- ❑ Log in to use:weblogic/weblogic.
- ❑ Click **Deployments**, check if SAPI State is **Active**.

For example:

Figure 44 SAPI Admin Console Window



Back Up and Restore

Backing Up Profile

Use `backupConfig.sh` to take a backup of the Profiles that are created using WebSphere.

Example:

```
[root@k8306-10 bin]# ./backupConfig.sh
ADMU0116I: Tool information is being logged in file
           /data/sapi/DMGR/logs/backupConfig.log
ADMU0128I: Starting tool with the SAPI_DGMR profile
ADMU5001I: Backing up config directory /data/sapi/DMGR/config to file
           /data/sapi/DMGR/bin/WebSphereConfig_2010-05-06.zip
ADMU0505I: Servers found in configuration:
ADMU0506I: Server name: dmgr
```

```
ADMU2010I: Stopping all server processes for node SAPI_DMGR_NODE
```

```
ADMU0510I: Server dmgr is now STOPPED
```

```
.....
.....
```

```
ADMU5002I: 1,475 files successfully backed up
```

Location of the file: Installed Directory/Domain Manager/bin/

Example: /data/sapi/DMGR/bin/backupConfig.sh

By default the backup stores the content in the same location. To specify a different location, change the target path in sapi-backup.cfg (chk.script.path = \${cbs.sapi.server.installdir}/bin/backup.sh) which is available in the Installed directory/server/config folder.

Example: /data/sapi/server/config/sapi-backup.cfg

Restoring Profile

Use restoreConfig.sh to restore the Profile from the backup Profile.

Location of the file: Installed Directory/Domain Manager/bin/

Example: /data/sapi/DMGR/bin/restoreConfig.sh

Backing Up Deployment Manager and Managed Servers

To back up Deployment Manager and Managed Servers for WebSphere, perform the following steps:

1. Run the backup job from UPM or run backup.sh.
/home/sapi_ws/server/bin SAPI
A zip file (/var/msf/transfers/backup/cbs_<component>_backup-\${hostName}.zip) is created.
2. Transfer the zip file to the UPM for storage.

Restoring Deployment Manager and Managed Servers

To restore Deployment Manager and Managed Servers for WebSphere, perform the following steps:

1. Transfer the backup file to the host machine and save it under the root directory (/).
2. Run unzip o cbs_<component>_backup-\${hostName}.zip
cd /home/<component>_ws
Change Ownership
chown -R apiuser:users * ==== for SAPI



NOTE

While restoring Deployment Manager or Managed Server, the IP and the hostname must be the same as it was before. When a new box is used, the setup for restoring it must have the same IP and hostname. This is necessary for the network setup. The IP/hostname is already configured for the application.

3. Once the restoring process is complete, start the servers.

Chapter 16

ASU Process

16

orded (sender
The desti
notifying
ng The noti
ieve The m
cT access To

e

v

ASU Process Description

The Application Server Unit (ASU) provides the Billing IVR (Interactive Voice Response) capabilities. The major functionalities include Admin Menu selection, Query Balance Information, Recharge subscriber or account balances, and so on. The ASU-IVR supports localization of the currently supported languages by using the IVR Language Localization platform. The IVR language localization platform is designed as a Web UI application. The platform provides a unified framework for localization team to test, modify, and package localized IVR prompts.

Prerequisites for Running ASU Process

Before starting the IVR application in ASU, make sure that:

1. All the Rating DBs are up and running.
2. All the Billing DBs are up and running.
3. The catalog, unscaled DBs are up and running.

Running ASU Process

To start the IVR app in ASU through command line:

```
>> cd /home/BEA/user_projects/domains/ivr_app/bin
>> ./ivrStart.pl
```

Environmental Variables

The following environment variables should be exported:

CBS_IVR_SERVER_INSTALLDIR = \$DOMAIN_HOME

WL_SCRIPTS = \$DOMAIN_HOME/bin

System Parameters

SuspSubInfoPrmt

0 = Disable/Play regular InfoServAnnounc, 1 = Enable/Play SuspendSub InfoServAnnounc

Fast Recharge

0 = Fast Recharge is Disabled and Not Supported,

number other than 0=Fast Recharge is Enabled and the number provisioned is the Fast Recharge prefix.

BAL_PLAY_ALWAYS

For play zero non-core applicable balance(s), 0=OFF/DISABLED, 1=ON/ENABLED

UNIFIED_CARD_PREFIX

A prefix used by ASU-IVRU to route a "UNIFIED_CARD " type call to SLU

TELEPHONE_PREFIX

A prefix used by ASU-IVRU to route a "TELEPHONE" type call to SLU

CALLING_CARD_PREFIX

A prefix used by ASU-IVRU to route a "CALLING_CARD" type call to SLU

Reviewing ASU Process Activity



How to Collect Trace

1. Turn on the logs:

```
>> cd /home/BEA/user_projects/domains/ivr_app
>> vi log4j.xml
```

Modify the log4j.xml as following, (change error to debug)

```
<appendername="rolling_file_appender"
class="org.apache.log4j.RollingFileAppender">
  <param name="File" value="SingleAPI.log" />
  <param name="BufferedIO" value="true" />
  <param name="Append" value="true" />
  <param name="MaxFileSize" value="10MB" />
  <param name="MaxBackupIndex" value="50" />
  <param name="Threshold" value="debug" />
  <layout class="org.apache.log4j.PatternLayout">
    <param name="ConversionPattern" value="%d | %-5p | %c | %m | %t |
    %.4X{httpSID}-%.4X{sessionId} | %x %n"/>
  </layout>
</appender>
```

2. Collect the ASU trace.

```
>> cd /home/BEA/user_projects/domains/ivr_app/logs
>> > SingleAPI.log
```

Make a call.

The trace should be in SingleAPI.log, and save it for further analysis.

Troubleshooting Guidelines

Use webmonitor tool to monitor the health of the ASU

A webmonitor tool is developed to monitor the health of the ASU. Login to any terminal server, open a Web browser, and use following URL to monitor the ASU:

<http://<asu ip>:7001/webmonitor/monitor.jsp>

An example of the monitor page is as follows. It reports the database connection status, and also the caching status.

Figure 45 Example - Monitor Page

SAPI LBA Health Check Report - Microsoft Internet Explorer

Address: <http://10.230.14.121:7001/webmonitor/monitor.jsp>

ASU Health Check Report

(SAPI is deployed as CONVERGED)

Database Status :

Overall health : Good.

Database Type	Database Name	Server ID	Status	Error Message
BILLING_CATALOG	CBSCTLG	1	good	
BILLING_CUSTOMER	CBSCUST	3	good	
UNSCALEDPC	CBSBLUS	4	good	
RATING_MASTER	CBSMAN	9	good	
RATING_HISTORY	CBSHIST	9	good	

Cache Statistics(Current Service Version: 1)

Cache Name	Cache Size	Current Service Version Cache Size	Unit
mAnncTypePromptMap	50	50	Records
mLanguageMap	9	9	Records

Done Internet

Chapter 17

Operational UFP Utility Details

17

orded (sender
The desti
notifying
ng The noti
ieve The m
cT access To

e

v

UFP Utility Description

UFP is the Usage File Purge utility. UFP performs the following:

- Identify files tracked in USAGE_FILE_STATUS/USAGE_FILES_TO_PURGE that are ready to be purged.
- Connect to the appropriate machine via FTP/STFP as configured in HOST_CONTACTS
- Delete the files from the appropriate machine, then delete the USAGE_FILES_TO_PURGE entry.
- UFP connects to either the ORP or the BLUS (unscaled databases). It only deletes files tracked in the database it is running against. To purge all files in the system two UFP instances must be run one against ORP and one against BLUS.

Prerequisites for Running UFP Utility

The prerequisites are:

- Correct configuration of Environment variables.
- Correct configuration of a row in HOST_CONTACTS for the machine where this utility is running.
- Configuration of the database names in SERVER_DEFINITION.dsquery field in the ORP database. The database names there must appear in the TNS file.

Dependencies on Other Processes/Utilities

UFP is typically started by mshell. UFP requires UPA to be running unless the OAM_ENV_CONN_MA is set to *false*.

Running UFP Utility

Procedure to run the UFP utility is given below:

1. Run from UPM.

2. Run from command line:

```
UFP <process_instance_name> <orp or blus database server id>
```

Example:

```
UFP   ufp01   4
```

Where,

Process_Name: process name is recommended to start with 'ufp' and have a 2 digit number.

Orp or blus database server_id – The server id of any server in the SERVER_DEFINITION table with a server_type of 4 or 12.

Configuration of UFP Utility

\$TNS_ADMIN/tnsnames.ora must have appropriate entries for accessing rating DB and billing DB (unscaled and customer db).

No other configuration file is needed. All configurations are done in the database or in the Environment variables.

Environmental Variables

The environmental variables are:

- ARBORLOG: Path to where the log file should go to (i.e. /staging/billing/log).
- ARBORCTRLRPT: Path to where the control report file should go to (i.e. /staging/billing/reports).
- ARBORBIN: Where all the binaries located.
- ARBOR_CATALOG_QUERY: Catalog database name in the TNS file - used to get the SERVER_DEFINITION table for the server_id values and dsquery names.
- ARBORDBU: User schema to log in database (i.e. cbs_owner) used only if CBS_USER not populated.
- TNS_ADMIN: Show the position where tnsnames.ora locates in.
- OAM_ENV_CONN_MA :
 - = TRUE (Connect to UPA for process monitoring during start up)
 - = FALSE (Do NOT connect to UPA for process monitoring)

System Parameters

Table 15 System Parameters

Module	Parameter Name	Default	Use
UFP	LOGLEVEL	MEDIUM	Configure modules to log errors of a designated severity level or higher. There are three levels of log detail: LOW - logs ERROR, ALARM, and FATAL messages MEDIUM - logs INFORM, WARNING, ALARM, ERROR, and FATAL messages HIGH - logs all messages and log SQL directed to this log.
UFP	TRA_SWITCH	0	SQL log files are written to the \$ARBORLOG directory 0 - No SQL logging 1 - Detailed SQL logging. One log file is created for each database connection in used. 2 - Detailed SQL logging. All SQL Written to a single SQL log file. 3 - Detailed SQL logging. One SQL log is created for each target DB. 4 - Detailed SQL logging. Written to the standard log file. Logging entries can be truncated.
UFP	REPORT_LEVEL	MEDIUM	Controls the content in the various report files and configuration logs: NO_DEBUG - FATAL content only LOW - add errors LOW_MED - add warnings MEDIUM - add informational HIGH_MED - add debug HIGH - add trace content

Table 15 System Parameters

Module	Parameter Name	Default	Use
UFP	ACTIVATE_TIMERS	0	Not relevant to these processes but reported in the configuration report.
UFP	WAKE_UP_INTERVAL_MINUTES	10	The number of minutes UFP waits after completing all current purges, before looking for more files to purge
UFP	ACTION_NODES	1	Number of threads performing the action of the process (purge or replication)
UFP	ACTION_QUEUE_SIZE	10	Number of files that are queued up waiting for processing
UFP	MAX_HUNG_PROCESS_MIN	20	Number of minutes the process waits before attempting to self terminate due to a hung thread. Range 10-120
UFP	MAX_RPT_FILE_ROLL	10	Maximum Number of report and log files per report after size is exceeded. Range 1-10,000
UFP	MAX_RPT_FILE_SIZE_MB	10	Number of megabytes per report and log files before the file rolls. Range 1-1024 (1MB-1GB)

HOST_CONTACTS

The machine where UFP is run, there should be a HOST_CONTACTS entry configured for that machine.

The HOST_CONTACTS entry for UFP may have any value for HOST_MODULE.

The processes look for any matching entry for its current host.

HOST_CONTACTS rows MUST be configured the same in both the ORP and the UNSCALED databases.

Optimal UFP Utility Schedule

UFP runs continuously. It is not expected that more than two UFP processes need to be run in each deployment. One for BLUS and one for ORP server.

Reviewing UFP Utility Activity

Log file and configuration report located under \$ARBORLOG directory.

Example of log file name convention:

```
ufp01-090422-045516-04-1306862.BATCH
```

```
ufp1-090422-045516-04-1306862.cfg
```

The .BATCH file is the log. If the log file exceeds 10MB then it is renamed to <log>.1 through <log>.10. The current log never has a number after .BATCH and the oldest log has .BATCH.10. After .10 is reached, the log is deleted and only the most recent 10 logs are kept.

The cfg file is the configuration report, a report that contains the configuration that was read from the DATABASE when the process ran.

The Control report of process activity is located under \$ ARBORCTRLRPT directory.

Example of log file name convention:

ufp01-090422-051355-04-1146966.rpt

If the ARBORCTRLRPT directory is not specified then the report is placed in the \$ARBORLOG directory.

Process activity is reported in BOTH the log and the Control report.

A status report of the process can be created by sending a `kill -USR2 <pid>`. The status report reports recent activity, and current thread status. This report ends with .status and is written to the \$ARBORLOG file.

Configuration reports are renamed daily to have (process_name>_LASTRUN_DAY_X.extension where X is number of the day of the week the logs was started. This renaming occurs on normal process termination or shortly after midnight each day. So when all is operating normally, logs are available for the last 7 days. Logs from situations where the process fails for some reason are not renamed and are kept for 30 days.

Expected Input for UFP Utility

Rows in the USAGE_FILES_TO_PURGE table.

Expected Output and Location of Output

No output is expected.

List of Success/Failure Criteria for UFP Utility

Review purged file to see that files are deleted.

Troubleshooting Guidelines

Start with the LOG and CFG files. There is a clear statement about the problem, any alarms raised and the configuration setting at the time the process ran.

Two Major categories: Startup and Operational problems

Startup:

The log file reports any configuration problems that are preventing operation. Examine the log file for details of why the process failed.

Operational problems:

Operational problems that identify a problem with the system during runtime appear in the process logs.

If UFP failed due to an unrecoverable error during operation, the fastest means to locate that error is to search for SignalS in the log. The cause of the error appears immediately above the message with SignalShutdown in the message.

Maintenance Guidelines

The maintenance guidelines are given below:

File System Maintenance

Log files and Configuration reports are renamed daily to form a rolling backup of the last 7 days of logs. Configuration reports from processes that end in failure are aged out after 30 days.

Database Maintenance

The process performs its own database maintenance.

Chapter 18

Operational UFR Utility Details

18

orded (sender
The desti
notifying
ng The noti
ieve The m
cT access To

e

v

UFR Utility Description

UFR is the Usage File Replication utility. UFR performs the following:

- Identify files tracked in USAGE_FILE_STATUS/USAGE_FILES_TO_REPLICATE tables that are ready to be replicated. These files must have a replication target on the MACHINE where UFR is running.
- Connect to the appropriate source machine via FTP/SFTP as configured in HOST_CONTACTS.
- Pull/replicate the file via SFTP/FTP to the target machine.
- Delete the USAGE_FILES_TO_REPLICATE entry and insert a row into USAGE_FILES_TO_PURGE so that the replicate file also purges.
- UFR connects to either the ORP or the BLUS (unscaled databases). It only replicates files tracked in the database it is running against. To replicate files tracked in both databases to the same target machine two UFR instances must be run: one against ORP and one against BLUS.

Prerequisites for Running UFR Utility

The prerequisites are:

- Correct configuration of Environment variables.
- Correct configuration of a row in HOST_CONTACTS for the machine where this utility is running.
- Configuration of the database names in SERVER_DEFINITION.dsquery field in the ORP database. The database names must appear in the TNS file.

Dependencies on Other Processes/Utilities

UFR is typically started by mshell. UFR requires UPA to be running unless the OAM_ENV_CONN_MA is set to *false*.

Running UFR Utility

The procedure to run the UFR utility is given below:

1. Run from UPM
2. Run from command line:

```
UFR <process_instance_name> <orp or blus database server id>
```

Example:

```
UFR   ufr01   4
```

Where,

Process_Name: process name is recommended to start with 'ufr' and have a 2 digit number.

Orp or blus database server_id: The server id of any server in the SERVER_DEFINITION table with a server_type of 4 or 12.

Configuration of UFR Utility

\$TNS_ADMIN/tnsnames.ora must have appropriate entries for accessing rating DB and billing DB (unscaled and customer db).

No other configuration file is needed. All configurations are done in the database or in the Environment variables.

Environmental Variables

The environmental variables are:

- ARBORLOG: Path to where the log file should go to (i.e. /staging/billing/log)
- ARBORCTRLRPT: Path to where the control report file should go to (i.e. /staging/billing/reports)
- ARBORBIN : Where all the binaries located
- ARBOR_CATALOG_QUERY: Catalog database name in the TNS file - used to get the SERVER_DEFINITION table for the server_id values and dsquery names.
- ARBORDBU: User schema to log in database (i.e. cbs_owner) used only if CBS_USER not populated
- TNS_ADMIN: Show the position where tnsnames.ora locates in.
- OAM_ENV_CONN_MA :
 - = TRUE (Connect to UPA for process monitoring during start up)
 - = FALSE (Do NOT connect to UPA for process monitoring)
- UFR_AUTOEXIT: Value used when testing the system. This value controls how many files are replicated before process termination. Negative -1 process all files currently waiting for replication then terminates.

System Parameters

Table 16 System Parameters

Module	Parameter Name	Default	Use
UFR	LOGLEVEL	MEDIUM	Configure modules to log errors of a designated severity level or higher. There are three levels of log detail: LOW - logs ERROR, ALARM, and FATAL messages MEDIUM - logs INFORM, WARNING, ALARM, ERROR, and FATAL messages HIGH - logs all messages and log SQL directed to this log.
UFR	TRA_SWITCH	0	SQL log files are written to the \$ARBORLOG directory 0 - No SQL logging 1 - Detailed SQL logging. One log file is created for each database connection in used. 2 - Detailed SQL logging. All SQL Written to a single SQL log file. 3 - Detailed SQL logging. One SQL log is created for each target DB. 4 - Detailed SQL logging. Written to the standard log file. Logging entries can be truncated.

Table 16 System Parameters

Module	Parameter Name	Default	Use
UFR	REPORT_LEVEL	MEDIUM	Controls the content in the various report files and configuration logs: NO_DEBUG - FATAL content only LOW - add errors LOW_MED - add warnings MEDIUM - add informational HIGH_MED - add debug HIGH - add trace content
UFR	ACTIVATE_TIMERS	0	Not relevant to these processes but reported in the configuration report.
UFR	WAKE_UP_INTERVAL	10	The number of seconds UFR waits before looking for more files to replicate.
UFR	ACTION_NODES	1	Number of threads performing the action of the process (purge or replication)
UFR	ACTION_QUEUE_SIZE	10	Number of files that are queued up waiting for processing
UFR	MAX_HUNG_PROCESS_MIN	20	Number of minutes the process waits before attempting to self terminate due to a hung thread. Range 10-120
UFR	MAX_RPT_FILE_ROLL	10	Maximum Number of report and log files per report after size is exceeded. Range 1-10,000
UFR	MAX_RPT_FILE_SIZE_MB	1	Number of megabytes per report and log files before the file rolls. Range 1-1024 (1MB-1GB)

HOST_CONTACTS

The machine where UFR is run, there has to be a HOST_CONTACTS entry configured for that machine. UFR pulls content to the machine it is running on. Therefore UFR identifies all HOST_CONTACTS entries that are TARGETS on the current host and it performs all replication for the host machine it is currently running on.

The HOST_CONTACTS entry for UFR may have any value for HOST_MODULE:

The processes look for any matching entry for its current host.

HOST_CONTACTS rows MUST be configured the same in both the ORP and the UNSCALED databases.



To configure HOST_CONTACTS for Replication

1. Define the host_contacts row for the TARGET machine
2. Define the host_contacts row for the SOURCE machine
3. On the SOURCE host contacts row set the replication_type = 2

4. On the SOURCE host contacts row set the replication_host_contact_id = host_contact_id of the host_contacts row of the TARGET machine.
5. Limitations:
 - ❑ It is safe to have multiple HOST_CONTACTS rows have the same REPLICATION_HOST_CONTACT_ID.
 - ❑ It is NOT ok to have a HOST_CONTACTS row where host_contact_id= replication_host_contact_id (replication to itself).

Optimal UFR Utility Schedule

UFR runs continuously.

ONE UFR process must be run on EACH target machine for each of the databases (ORP and BLUS) where files need replication.

UFR reads from the USAGE_FILES.

It is not expected that more than two UFR processes need to be run on each replication target machine in a deployment. One for BLUS and one for ORP server per target machine.

Reviewing UFR Utility Activity

Log file and configuration report located under \$ARBORLOG directory.

Example of log file name convention:

```
ufr01-090422-045516-04-1306862.BATCH
```

```
ufr1-090422-045516-04-1306862.cfg
```

The .BATCH file is the log. If the log file exceeds 10MB it is renamed to <log>.1 through <log>.10. The current log never has a number after .BATCH and the oldest log has .BATCH.10. After .10 is reached the log are deleted and only the most recent 10 logs are kept.

The cfg file is the configuration report, a report that contains the configuration that was read from the DATABASE when the process ran.

The Control report of process activity is located under \$ ARBORCTRLRPT directory.

Example of log file name convention:

```
ufr01-090422-051355-04-1146966.rpt
```

If the ARBORCTRLRPT directory is not be specified then the report is placed in the \$ARBORLOG directory.

Process activity is reported in BOTH the log and the Control report.

A status report of the process can be created by sending a `kill -USR2 <pid>`. The status report reports recent activity, and current thread status. This report ends with .status and is written to the \$ARBORLOG file.

Configuration reports are renamed daily to have <process_name>_LASTRUN_DAY_X.extension where X is number of the day of the week the logs was started. This renaming occurs on normal process termination or shortly after midnight each day. So when all is operating normally the logs are available for the last 7 days. Logs from situations where the process fails for some reason are not renamed and are kept for 30 days.

Expected Input for UFR Utility

Rows in the USAGE_FILES_TO_REPLICATE table.

Expected Output and Location of Output

No output is expected.

List of Success/Failure Criteria for UFR Utility

USAGE_FILE_STATUS.replication_status is set to 2 – indicating replication completed and the USAGE_FILES_TO_REPLICATE row is deleted.

Troubleshooting Guidelines

Start with the LOG and CFG files. There is a clear statement about the problem, any alarms raised and the configuration setting at the time the process ran.

Two Major categories: Startup and Operational problems

Startup:

The log file reports any configuration problems that are preventing operation. Examine the log file for details of why the process failed.

Operational problems:

Operational problems that identify a problem with the system during runtime appear in the process logs.

If UFR failed due to an unrecoverable error during operation, the fastest means to locate that error is to search for SignalS in the log. The cause of the error appears immediately above the message with SignalShutdown in the message.

Maintenance Guidelines

The maintenance guidelines are given below:

File System Maintenance

Log files and Configuration reports are renamed daily to form a rolling backup of the last 7 days of logs. Configuration reports from processes that end in failure are aged out after 30 days.

Database Maintenance

The process performs its own database maintenance.

Chapter 19

CTI Functionality

19

orded (sender
The desti
notifying
ng The noti
ieve The m
cT access To

e

v

CTI Functionality Description

Comverse provides an interface by which a small set of specific Customer Center screens can be popped from within a Comverse customer's third-party application. This chapter describes the procedures for setting up this interface.

Prerequisites for Running CTI Functionality

Before the CTI functionality can be run successfully, the `FXFrameConfiguration.properties` file must enable the CTI functionality as described below. The `system_parameters` table settings must also be verified for proper configuration.

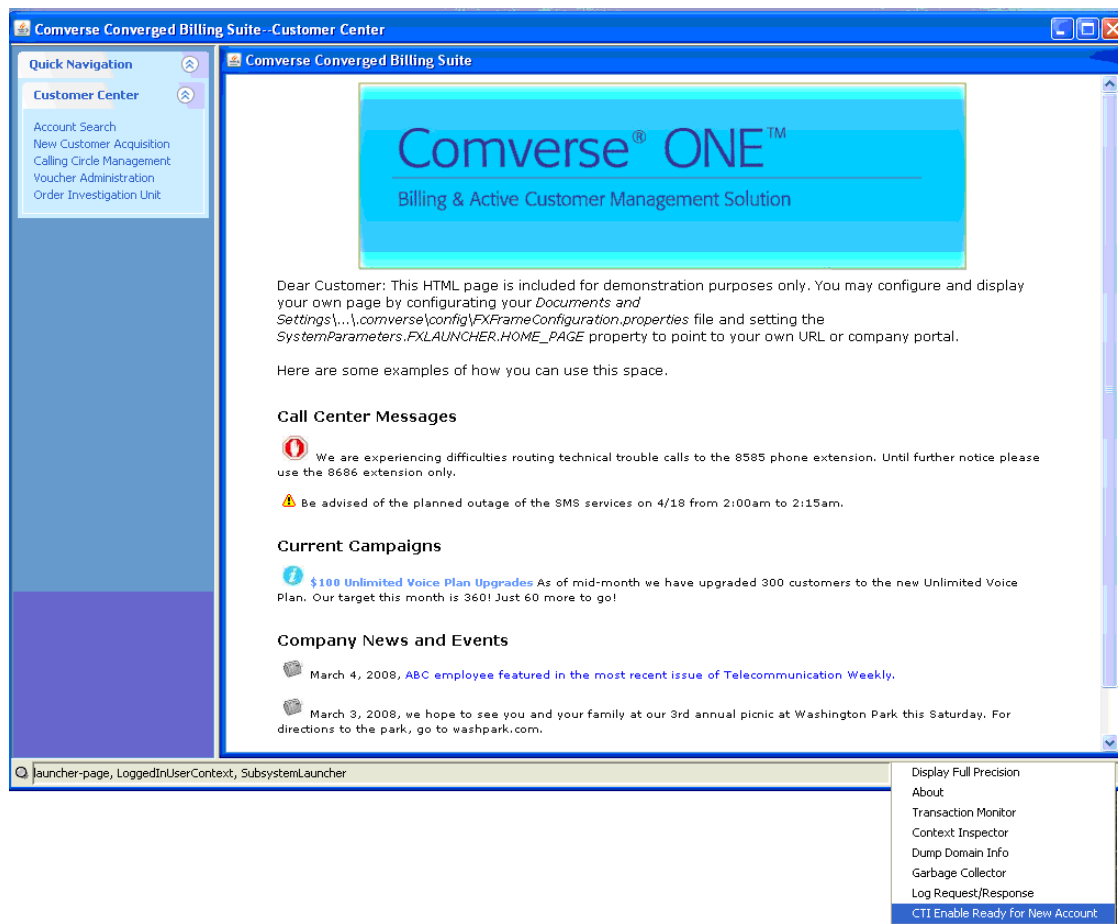
Dependencies on Other Processes/Utilities

There are no dependencies on other CSM processes or utilities.

Running CTI

The CTI functionality, when enabled, pops the Account number when it receives an Account number via the socket interface at the specified host and port defined by the `system_parameters` table. When the user has finished viewing the Account and the End Session action is initiated, the user must then select the menu item to indicate that another Account number can now be processed.

The figure below shows the menu item that is used to indicate a Ready or Busy state for the user. When the menu item is in the Checked state, this indicates a Ready state. The Ready state must be set manually by the user when a new Account is ready to be processed.

Figure 46 Menu Item - Indicate Ready or Busy

The following steps describe the actions to run the CTI functionality within the CSM application.

1. Start CSM application.
2. Login to CSM application.
3. Verify CSM application menu item is showing that it is in a Ready state. If not, be sure to set the CSM application into the Ready state. Verify that the menu item is in a checked state.
4. Select Account Search action link from CSM Customer Center navigation pane.
5. The CSM application navigates to the appropriate Account Summary panel or the AccountLocate panel.
6. If Account Summary panel is displayed, select the End Session link when done viewing the Account.
7. Set the CTI Menu Item to a Ready State. Verify that the menu item is in a checked state.
8. Proceed to step #5.

Environmental Variables

The `FXFrameConfiguration.properties` file should contain the `cti.enable` parameter in order to enable CTI functionality. If it does not exist, or if it already exists, verify that it is enabled with a true value.

Here is an example of how to configure this setting:

```
cti.enable=true
```

The FXFrameConfiguration.properties that was edited needs to be placed in the appropriate directory. This file needs to be copied to the .comverse\config folder within the user folder.

For example, C:\Documents and Settings\<userid>\.comverse\config

If this directory does not exist within the user folder, you need to create a .comverse folder within the user folder. Within the .comverse folder you need to create a config folder. Pay special attention to the period preceding the work comverse when creating this directory. It must contain this period. Copy the changed FXFrameConfiguration.properties to this directory.

The FXFrameConfiguration.properties file may also be configured with the cti.sleep.time parameter. This is the configuration setting that controls the polling time in milliseconds for monitoring the SAPI functionality that retrieves the Account information from the socket connection.

Here is an example of how to configure this setting:

cti.sleep.time=4000

System Parameters

The following are default settings in the system_parameters table.

Table 17 System Parameters

Module	Parameter Name	Description
cti	TIME_TO_WAIT	Default value 3500. Polling time value for the SAPI socket connection interface in milliseconds. This can be adjusted as necessary to suite any network latencies.
cti	HOST_NAME	Default value of localhost. Contains the IP address for the host.
cti	PORT_NUMBER	Default value of 11095. Contains the value of the host port number.
cti	READY_SIGNAL	Default value of READY. Contains the signal to indicate a ready condition to the socket connection.
cti	USE_BUSY_SIGNAL	Default value of 1.
cti	BUSY_SIGNAL	Default value of BUSY. Contains the signal to indicate a busy condition to the socket connection.

Reviewing CTI Functionality

CTI functionality information logging is logged to the standard CSM log file for both the CSM GUI as well as the SAPI logic.

The following is the list of messages logged for various scenarios. All messages are logged in the FXLauncher.log file.

- CTI functionality not enabled (cti.enabled=false).
- CTI Not enabled.CtiStartMonitor action ignored

Account information specified did not locate an account in the system. This message is logged within the CSM.log file.

- No account found for account summary.Payload:<parameter list>

Invalid information received within socket message. This message is logged within the SAPI log file.

- No valid data received

Attribute received within the socket message is not a recognized attribute. This message is logged within the SAPI log file.

- Attribute name is not a recognized attribute
- <AccountExternalId/CustPhone1>

Exception related to socket processing. This message is logged within the SAPI log file.

- Exception:<socket processing exception>

Expected Input

Input for the CTI functionality is expected to come from an external source to the specified socket interface which indicates a ready, busy signal or an Account number.

The socket-based message is expected to be in the following format:

action=acct-summary/AccountExternalId=xyz" or "action=acct-summary/CustPhone1 = zabpq1234

Where,

- xyz is the Account External Id Identifier and
- zabpq1234 is the Service Instance External Id Identifier

Sample message:

action=acct-summary/AccountExternalId=258

The following messages are sent from the CSM application based on the respective action:

- READY
- BUSY

Each message is appended with `\r\n` to signify the end of message string.

List of Success/Failure Criteria for CTI Functionality

When the Account number has been successfully retrieved from the socket interface and processed appropriately, navigation changes from the Account Search panel to the Account Summary panel and the Account number information is displayed.

Troubleshooting Guidelines

If the behavior of the CTI functionality does not operate as expected, then the `system_parameter.TIME_TO_WAIT` parameter can be adjusted to avoid a potential timeout at the socket connection.

Chapter 20

OCS Process

20

orded (sender
The desti
notifying
ng The noti
ieve The m
cT access To

e

OCS Description

OCS is an implementation of the Diameter Credit Control Application (DCCA) as extended by the 3GPP to support its additional requirements. Comverse OCS is designed to integrate Diameter and 3GPP requirements with Comverse Converged Billing Suite functions. The functionality described by the 3GPP OCS is a natural extension of the capabilities of the Comverse Converged Billing Suite.

Dependencies on Other Processes

The following are the dependencies to make OCS process work correctly.

1. TSP
2. URE_Q, URE_U
3. tcpipdata

Running OCS Process

To run the OCS process, add the following to cestart.202 file:

```
OCS_SLF . 0 DEF DEF 202 100 /home/omni/bin/ocs -config  
/home/omni/conf/ocs.xml
```

Configuration of OCS Process

To configure OCS, check the following:

1. Add the following TCPIP entry in the file /home/omni/conf/ nodeConf.IPC.202.
TCPIP_RP IPCTcpNotReady
2. Add the following TCPIP entry in the file /home/omni/conf/ nodestart.IP.202.
TCPIP_RP 1 DEF DEF 202 100 tcpipctrl
3. Add the following in tapdes.202 file(optional) to monitor the OCS process.
OCS_SLF
4. Modify cestart.202 and tnsnames.ora as per each status requirement. In cestart.202 file, add the following to make OCS process up.

```
OCS_SLF . 0 DEF DEF 202 100 /home/omni/bin/ocs -config  
/home/omni/conf/ocs.xml
```

Environmental Variables

None

System Parameters

OCS_CDR

OCS CDR enable = 1, disable = 0

OCS_ENABLED

Diameter OCS Possible Values: Enabled = 1, Disabled = 0

OCS_SUB_LOC

Diameter OCS default subscriber location

OCS_SUB_LOC_TYPE

Li Relation

Possible values: 1 (UNLISTED), 1 (HANDSET_MAP), 2 (TELEPHONE), 3 (MSRN), 4 (CELL_ID), 5 (MSC_ID), 6 (SGSN_ID), 7 (IP_ADDRESS)

OCS_OTH_LOC

Def Other location

OCS_OTH_LOC_TYPE

Li Relation, Possible values: 1 (UNLISTED), 1 (HANDSET_MAP), 2 (TELEPHONE), 3 (MSRN), 4 (CELL_ID), 5 (MSC_ID), 6 (SGSN_ID), 7 (IP_ADDRESS)

Optimal OCS Process Schedule

None

Reviewing OCS Process Activity

When the OCS and related process are started, a log file is generated and is stored at /home/omni/Logs. This contains the information about the process start and end status.

In directory /home/omni/<sluname>/tmp, Event.202.txt.* files are generated which contains the OCS related process activity information.

Expected Input for OCS Process

OCS process receives the request from the OCS client and processes it. OCS process receives the following requests.

- CCR-Initial
- CCR-Update
- CCR-Terminate
- CCR-Immed for DirectDebit Operartion
- CCR-Immed for Advice of Charge
- CCR-Refund

Expected Output and Location of Output

The following responses are sent to the OCS client for the appropriate request.

- CCR-Initial Response
- CCR-Update Response
- CCR-Terminate Response
- CCR-Immed Response (for DirectDebit Operartion)
- CCR-Immed Response (for Advice of Charge)
- CCR-Refund Response

List of Success/Failure Criteria for OCS Process

None

Troubleshooting Guidelines

The following configuration steps are followed for the process to be up and running.

1. omd OCS_SLF
2. OCS_SLF>>

Maintenance Guidelines

None

File System Maintenance

None

Database Maintenance

None

Chapter 21

OSA Process

21

orded (sender
The desti
notifying
ng The noti
ieve The m
cT access To

e

OSA Description

OSA process allows the operator to make real time charges to subscribers for enhanced services. Operators can now offer a wider range of features and capabilities to the prepaid subscriber, thus enhancing the subscriber's experience. The Parlay/OSA Charging API implements a strong set of features associated with the Converged Billing Suite, along with some additional rating capabilities. The operator has a wide range of flexibility to make charges based on algorithms using currency or unit based rating.

Dependencies on Other Processes

The following are the dependencies to make OSA process work correctly.

1. TSP
2. URE_Q,URE_U
3. tcpipdata

Running OSA Process

Add the following to cestart.202 file to run the OSA process:

```
SLF_OSA . 0 DEF DEF 202 100 /home/omni/bin/osasoap
```

Configuration of OSA Process

To configure OSA, check the following:

1. Add the following TCPIP entry in the file /home/omni/conf/ nodeConf.IPC.202 TCPIP.
TCPIP_RP IPCTcpNotReady
2. Add the following TCPIP entry in the file /home/omni/conf/ nodestart.IP.202.
TCPIP_RP 1 DEF DEF 202 100 tcpipctrl
3. Add the following entry in the tapdes.202 file(optional) to monitor the OSA process.
SLF_OSA
4. Modify cestart.202 and tnsnames.ora as per each status requirement. To bring up the OSA process, add the following entry in the cestart.202 file .
SLF_OSA . 0 DEF DEF 202 100 /home/omni/bin/osasoap

Environmental Variables

None

System Parameters

OSA_ENABLED

ON/Enabled = 1, OFF/Disabled = 0

OSA_CDR

OFF/Disabled = 0, ON/Enabled = 1

OSA_Sub_Lookup

MSISDN based subscriber lookup for OSA = 0, IMSI based subscriber lookup for OSA = 1

Optimal OSA Process Schedule

None

Reviewing OSA Process Activity

When the OSA and related process are started, a log file is generated and is stored at /home/omni/Logs. This contains the information about the process start and end status.

In directory /home/omni/<sluname>/tmp, Event.202.txt.* files are generated which contains the OSA related process activity information.

Expected Input Requests for OSA Process

OSA process receives the request from the OSA client and processes it. OSA process receives the following requests.

- Charging Session Creation Method: createChargingSession()
- Session Release Method: release()
- Reservation Lifetime Extension Method: extendLifeTimeReq()
- Amount-based Reservation Creation Method: reserveAmountReq()
- Reserved Amount Debit Method: debitAmountReq()
- Reservation Credit Amount Method: creditAmountReq()
- Unit-Based Reservation Method: reserveUnitReq()
- Unit-based Reservation Debit Method: debitUnitReq()
- Unit-based Reservation Credit Method: creditUnitReq()
- Immediate Charging Debit Method: directDebitUnitReq()
- Immediate Charging Debit Method: directDebitAmountReq()
- Subscriber Balance Credit Method: directCreditAmountReq()

Expected Output and Location of Output

The following responses are sent to the OSA client for the appropriate request:

- Charging Session Creation Method: createChargingSession()
- Session Release Method: release()
- Reservation Lifetime Extension Method: extendLifeTimeRes()
- Amount-based Reservation Creation Method: reserveAmountRes()
- Reserved Amount Debit Method: debitAmountRes()
- Reservation Credit Amount Method: creditAmountRes()
- Unit-Based Reservation Method: reserveUnitRes()
- Unit-based Reservation Debit Method: debitUnitRes()
- Unit-based Reservation Credit Method: creditUnitRes()
- Immediate Charging Debit Method: directDebitUnitRes()
- Immediate Charging Debit Method: directDebitAmountRes()

- Subscriber Balance Credit Method: directCreditAmountRes()

List of Success/Failure Criteria for OSA Process

None

Troubleshooting Guidelines

The following configuration steps are followed for the process to be up and running.

1. omd SLF_OSA
2. SLF_OSA>>

Maintenance Guidelines

None

File System Maintenance

None

Database Maintenance

None

Chapter 22

USSD Process

22

orded (sender
The desti
notifying
ng The noti
ieve The m
t access To

e

USSD Description

USSD process is responsible for handling the Unstructured Supplementary Services Data (USSD) protocol messages. We can provision two Comverse ONE solution features that are based on the Unstructured Supplementary Services Data (USSD) protocol; USSD Callback and USSD Recharge and Information Server.

USSD Callback

The USSD Callback feature enables the Comverse ONE solution to support subscribers roaming in Global System for Mobile Communication (GSM) networks that are not compliant with the CAMEL Phase 2 standard.

USSD Recharge and Information Server

The USSD server provides a recharge and information service to subscribers without the use of IVR sessions that have the potential to overload the network. By dialing configurable service codes or information strings, subscribers retrieve information about accounts via text messages on their handsets or recharge accounts using a recharge voucher. USSD transmits information or instructions over the Global System for Mobile Communication (GSM) network in a manner similar to that of the Short Message Service (SMS). Unlike SMS, USSD is session-oriented and does not store and forward the information. When a user accesses a USSD service, a session is established and the radio connection stays open until the user or application terminates it. USSD commands are routed back to the home mobile network's Home Location Register (HLR) and work just as well when the users are roaming. When the service code entered by the subscriber is interpreted and validated, the system is able to retrieve account information such as balance and expiration date, or allow the subscriber to recharge an account balance using a recharge voucher.

Prerequisites for Running USSD Process

The detailed steps for the scratch installation are given below:

1. Install OS.
2. After the installation of OS, reboot the machine and copy the RPM's mentioned in the Installation Media/Package details section (step #2) to /data/install directory. Create the directory /data/install if it does not exist.
3. Make sure 3p entities are installed which are mentioned in Install 3P step.
4. Install RPMs. USSD process is a part of installer for CBS_SLU_BASE-3.0.0. Rating_Common-3.0.0 must be installed before Rating_USSD-3.0.0.
5. Run configureSLU.
6. Configure security server.
 - a. Install the SEC_API rpm.
 - b. Add security server IP to /etc/hosts, if it is not already there, with hostname secserv (the IP needs to be the actual IP of the security server).
 - c. Use the following parameters to provision an entry on the security server for Rating as well as Billing DB:
Username: <the User name for DB>
Password: <the Password for DB>
Default DB Instance: <instance_name>
Default DB Type: <DB_Type>

- d. Make sure the XML configuration file `commonCfg.xml` under `/home/omni/conf` directory contain the actual user name of the Rating DB in the `DefaultOracleUserName` tag.

Running USSD Process

To start OMNI, perform these actions:

1. Log in as **root**, using su password **sonora**.
2. Start OMNI with the `/sbin/service omni start` command.
3. To stop OMNI, use the command, `/sbin/service omni stop`.

Configuration of USSD Process

The detailed steps for the configuration of USSD are given below:

1. Modify `cestart.202` and `tnsnames.ora` as per each setups requirement. A `cestart.202` file is shown below:

```
OPPS . 1 DEF 202 100 /home/omni/bin/CallProcessor -name OPPS -T
TPPS . 1 DEF 202 100 /home/omni/bin/CallProcessor -tpps -name TPPS -T
PMTSVR . 1 DEF DEF 202 101 /home/omni/bin/PmtSvr -d -hbt 180
URE_Q1 . 1 DEF DEF 202 100 /home/omni/bin/ure -Q -ratingdb 1
URE_U1 . 2 DEF DEF 202 100 /home/omni/bin/ure -U -ratingdb 1
URE_Q9 . 1 DEF DEF 202 100 /home/omni/bin/ure -Q -ratingdb 9
URE_U9 . 2 DEF DEF 202 100 /home/omni/bin/ure -U -ratingdb 9
TSP . 1 DEF DEF 202 100 /home/omni/bin/tsp
USSD_SLF . 0 DEF DEF 202 100 /home/omni/bin/ussd_slf
SLF_SMS . 1 DEF DEF 202 100 /home/omni/bin/cap3sms
```

1. The following USSD configuration files are installed as part of the SLF rpm(s) (all in `/home/omni/conf`):

```
commonCfg.xml
ussdCfg.xml
```

Environmental Variables

The required environment variables are created while running `configureSLU`. You may edit `/home/sncpuser/profile/omnisetup.sh` to set environment variables appropriately.

System Parameters

```
MI_USSD_SWITCH
MI_FEATREQ_SWIT
USSDCB_BNUM_LKP
BAL_TRANSFER_ENABLED
DEPLOYMENT_MODE
NP_INTRFACE_TYP
```


USE_GLOBAL_USSD_RESP
SEND_DEFAULT_USSD_RESPONSE
NP_ENABLED
NP_ORIG_B_LOOKUP
REVENUE_REC_TYPE

Optimal USSD Process Schedule

None

Reviewing USSD Process Activity

When the USSD and related process are started, a log file is generated and is stored at /home/omni/Logs. This contains the information about the process start and end status.

Expected Input Requests for USSD Process

- UssdMessage
- WinFeatureRequestInd
- ResourceClear
- CallInfoFromResource
- RoutingInfoFromResource

Expected Output and Location of Output

- UssdMessage
- WinFeatureRequestResp

List of Success/Failure Criteria for USSD Process

The **clear_cause** field in CDR record and Usage Record (generated in usage_record_main table) determines the success/failure of the activity (16 for success and any other value for failure).

Troubleshooting Guidelines

None

Maintenance Guidelines

None

File System Maintenance

None

Database Maintenance

None

Chapter 23

Camel 3 GPRS Process

23

orded (sender
The desti
notifying
ng The noti
ieve The m
cT access To

e

v

Camel 3 GPRS Application Description

GPRS process is responsible for billing the Third Generation (3G) network packet data services using the CAMEL 3 General Packet Radio Services (GPRS) in GSM networks. GSM data services vary from streaming media to Internet Web surfing. These data services are fundamentally limited only by mobile handset capabilities and interface bandwidth.

The CAMEL 3 (CAP Phase 3) protocol has support for prepaid data services billing built directly into the protocol so that an SCP (such as the Converse ONE solution) can allow or disallow and charge for data transactions. GPRS process receives usage information for subscriber services in the form of octets or seconds from the GPRS network. This process rates this usage by converting it into corresponding currency units such as dollars.

The reservation-based mechanism is used for charging. A certain amount of usage is authorized and the corresponding balances are set aside when the network requests a reservation. The balances are permanently reduced only after consumption is confirmed. If a part of the reservation is not used, then it is returned and the balances are readjusted accordingly. Reservation amounts are configurable.

Dependencies on Other Processes

GPRS needs to register to TSP to get static data loaded.

Running Camel 3 GPRS Process

To start OMNI, perform these actions:

1. Log in as **root**, using su password **sonora**.
2. Start OMNI with the `/sbin/service omni start` command.
3. To stop OMNI, use the command, `/sbin/service omni stop`.

Configuration of Camel 3 GPRS Process

The detailed steps for the configuration of GPRS are given below:

1. Modify `cestart.202` and `tnsnames.ora` as per each setups requirement. A typical `cestart.202` file is shown below:

```
SLF_GPRS . 1 DEF DEF 202 101 /home/omni/bin/slf_gprs
```

The option `-max_contexts nnn` can be supplied on command-line, where `nnn` is the max number of call contexts. If not specified, max call context is defaulted to 12000.

The rest of configuration should be the same as all other applications (corresponding IPF process, IND, needs to be run; SLF_GPRS and IND both need to be included in `tapdes.202` and so on)

2. The following GPRS configuration files are installed as part of the SLF rpm(s) (all in `/home/omni/conf`):

```
commonCfg.xml
```

```
slf_gprsCfg.xml
```

```
tspCfg.xml
```

Environmental Variables

The required environment variables are created while running configureSLU. You may edit /home/sncpuser/profile/omnisetup.sh to set environment variables appropriately.

System Parameters

Set INT_VALUE for the RTNG/GPRS_CDR parameter to 1 if RTB style CDR file needs to be generated for a GPRS call. The default value is 0.

Optimal GPRS Process Schedule

None

Reviewing GPRS Process Activity

When the GPRS and related process are started, a log file is generated and is stored at /home/omni/Logs. This contains the information about the process start and end status.

Below configuration/settings can be verified at any point by using

```
SLF_GPRS>> SLF_GPRS,<OPTIONS>;
```

OPTIONS

HELP

CMPIL-DNMOD-SCRIPTS-PMTS

RESET-MEAS

DUMP-MEAS

DISPLAY-PMTS:ALL

DISPLAY-CFG

CHANGE-PMTS:THRESHOLD-TIMER=NN

CHANGE-PMTS:MSGs-PER-SEC=NN

DUMP-SVC-REC

DUMP-TRANS-BTREE

ENABLE-MODULE=<ALL> or <MODULE_NAME>

DISABLE-MODULE=<ALL> or <MODULE_NAME>

ENABLE-MODULE-MONITORING

DISABLE-MODULE-MONITORING

DISPLAY-MODULE-MONITORING-STATUS

DUMP-MODULE-LIST

DUMP-MODULE-INVOCATION-DATA

DUMP-MODULE-STATS

ENABLE-MODULE-STATS

DISABLE-MODULE-STATS

SUB=<nnn>

DISABLES

Expected Input for GPRS Process

GPRS process receives the request from the IND and processes it. GPRS process receives the following requests:

- Provide Instruction (PI)
- Transaction info from resource (TIFR)
- Resource Clear (RC)

Expected Output and Location of Output

The following response is sent to the IND for the appropriate request.

- Transaction Info to resource

List of Success/Failure Criteria for GPRS Process

The **clear_cause** field in CDR record and Usage Record (generated in usage_record_main table) determines the success/failure of the activity (16 for success and any other value for failure).

Troubleshooting Guidelines

None

Maintenance Guidelines

None

File System Maintenance

None

Database Maintenance

None

Chapter 24

ECI Process

24

orded (sender
The desti
notifying
ng The noti
ieve The m
t access To



ECI Process Description

The Comverse Converged Billing Suite Event Charging Interface is a TCP/IP-based interface enabling external services to support real-time authorization, rating, and charging for transactional usage. The Event Charging Interface can be used for charging many kinds of value-added services including sending and receiving SMS and MMS messages, Internet access and other data content, vending machine purchases, and various other transactions.

ECI supports two types of charging, the first type of charging allows the external service to debit a specific amount. This direct charge might be typical of a vending machine purchase and the second uses tariff information stored within the Converged Billing Suite system's Rating Engine. This tariff-based charge might be typical of an SMSC, which does not normally have rating capabilities.

Any charge can be backed-out or reversed, if the external service decides it is appropriate (that is, an SMS message was not successfully delivered).

Dependencies on Other Processes/Utilities

To make ECI process work correctly, following processes are the dependencies:

1. TSP
2. URE_Q, URE_U
3. tcpipdata

Running ECI Process Utility

To run the ECI process following line has to be added in to cestart.202 file:

```
PMTSVR . 1 DEF DEF 202 101 /home/omni/bin/PmtSvr -d -hbt 180
```

Configuration of ECI Process

To configure ECI check for the following given below:

1. In file /home/omni/conf/ nodeConf.IPC.202 there should be an entry of TCPIP
TCPIP_RP IPCtcpNotReady
2. In file /home/omni/conf/ nodestart.IP.202 there should be entry of TCPIP
TCPIP_RP 1 DEF DEF 202 100 tcpipctrl
3. To monitor the ECI process we can have a entry in tapdes.202 file(optional)
PMTSVR

System Parameters

Set INT_VALUE for the RTNG/PMT_CDR parameter to 1 if RTB style CDR file needs to be generated for a ECI call. The default value is 0.

Reviewing ECI Process Activity

When the ECI and related process are started, a log file is generated and stored at /home/omni/Logs. This contains the information about the process starts and end status.

Logs file particular to ECI are generated at:
/home/omni/<SLUNAME>/tmp/PMTSVR.202.dump if we dump the configuration/setting for the process.

Below configuration/settings can be verified at any point by using:

PMTSVR>> PMTSVR,<OPTIONS>;

OPTIONS:

HELP

CMPIL-DNMOD-SCRIPTS-PMTS

RESET-MEAS

DUMP-MEAS

DISPLAY-PMTS:ALL

DISPLAY-CFG

CHANGE-PMTS:THRESHOLD-TIMER=NN

CHANGE-PMTS:MSGs-PER-SEC=NN

DUMP-SVC-REC

DUMP-TRANS-BTREE

ENABLE-MODULE=<ALL> or <MODULE_NAME>

DISABLE-MODULE=<ALL> or <MODULE_NAME>

ENABLE-MODULE-MONITORING

DISABLE-MODULE-MONITORING

DISPLAY-MODULE-MONITORING-STATUS

DUMP-MODULE-LIST

DUMP-MODULE-INVOCATION-DATA

DUMP-MODULE-STATS

ENABLE-MODULE-STATS

DISABLE-MODULE-STATS

SUB=<nnn>

DISABLES

Expected Input for ECI Process

ECI process receives the request from the SMSC client and processes it. The list of requests received by ECI process is given below:

- Heartbeat Request
- Validate Subscriber Request
- Apply Charge Request
- Apply Tariff Request
- Transaction ID Acknowledge
- Reverse Charge

Expected Output and Location of Output

The list of response that are sent to the SMSC client for the appropriate request is given below.

- Heartbeat Response
- Validate Subscriber Response
- Apply Charge Response
- Apply Tariff Response
- Reverse Charge Response
- Invalid Message Response

List of Success/Failure Criteria for ECI Process

1. The table PMTSVR_SERVICE_MAPPING which contains fields Bearer_capability, Discount and External_unit_type_id determines success/failure. It should have a mapping of above fields to derive appropriate AUT_ID for particular transaction.
2. Check for the IP address and port at which ECI process is receiving the messages.

Troubleshooting Guidelines

As part of troubleshooting, if the above configuration steps are followed, the process will be up and running. If the ECI process is not receiving any message then check for the throttle limit.

Throttle limit can be verified by:

```
PMTSVR>>pmtsvr,DISPLAY-PMTS:ALL
```

MML sent to PMTSVR:

```
DISPLAY-PMTS:ALL
```

MML_RESP:

Messages/Second=300

Throttle Time=60

Received Messages=0

Processed Messages=0

Dropped Messages=0

To set the throttle limit as per requirement using:

```
CHANGE-PMTS:THRESHOLD-TIMER=NN
```

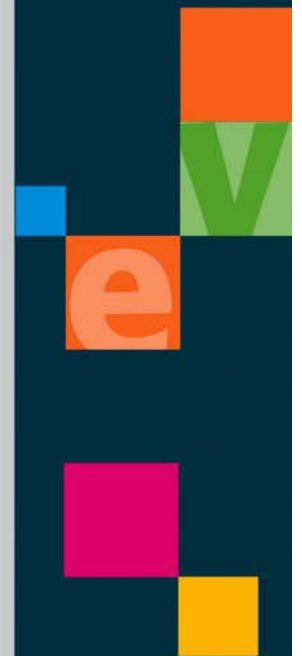
```
CHANGE-PMTS:MSGs-PER-SEC=NN
```


Chapter 25

CallProcessor Process

25

orded (sender
The desti
notifying
ng The noti
ieve The m
cT access To



CallProcessor Process Description

CallProcessor is the process responsible for handling all the voice calls. CallProcessor can run in two modes:

- OPPS
- TPPS

CallProcessor is event driven state model and it works in conjunction with the other SLF processes namely TSP and URE.

Detects the receipt of A1129 message from IPF layer and sends response or request to IPF layer. Determines on which SDP the subscriber resides by looking into the range_map. Sends an account request to URE_Q[n] process to fetch calling party information. If ACCOUNT_VALID check is done by URE for the subscriber validity(state). Resolves the subscriber's locations to get whether the call is roaming call or home location call. Pps_lbb_precedence is used to check the priority of resolving. Depending on the called number (if translated to a different number) sends a continue or connect. To bridge the call. If the call is answered then a charging API from URE is invoked and reservation requests are sent to SDS process in the state of MonitorAndBilling. When the call terminates then CDRs(files) are generated by the CallProcessor.

Dependencies on Other Processes

CallProcessor is dependent on RechargeAgent, IVRU, URE, and IP_INA process to communicate.

CallProcessor needs to register to TSP to get static data loaded.

Running CallProcessor Process

To start OMNI, perform these actions:

1. Log in as root, using su password sonora
2. Start OMNI with the /sbin/service omni start command.

To stop OMNI, use this command: /sbin/service omni stop

Configuration of CallProcessor Process

The detailed steps for the configuration of URE are given below:

1. Modify cestart.202 and tnsnames.ora as per each setups requirement. A typical cestart.202 file is shown below:


```

OPPS . 1 DEF DEF 202 100 /home/omni/bin/CallProcessor -name OPPS -T
TPPS . 1 DEF DEF 202 100 /home/omni/bin/CallProcessor -tpps -name TPPS -T
PMTSVR . 1 DEF DEF 202 101 /home/omni/bin/PmtSvr -d -hbt 180
URE_Q1 . 1 DEF DEF 202 100 /home/omni/bin/ure -Q -ratingdb 1
URE_U1 . 2 DEF DEF 202 100 /home/omni/bin/ure -U -ratingdb 1
URE_Q9 . 1 DEF DEF 202 100 /home/omni/bin/ure -Q -ratingdb 9
URE_U9 . 2 DEF DEF 202 100 /home/omni/bin/ure -U -ratingdb 9
TSP . 1 DEF DEF 202 100 /home/omni/bin/tsp
USSD_SLF . 0 DEF DEF 202 100 /home/omni/bin/ussd_slf
SLF_SMS . 1 DEF DEF 202 100 /home/omni/bin/cap3sms
      
```

2. The following URE configuration files are installed as part of the SLF rpm(s) (all in /home/omni/conf):
commonCfg.xml
CallProcessorCfg.xml

Environmental Variables

The environmental variables are:

The required environment variables are created while running configureSLU. You can edit /home/sncpuser/profile/omnisetup.sh to set environmental variables as appropriate.

System Parameters

FAST_RECHARGE
NP_INTRFACE_TYP
NP_ORIG_REDIRECT
MULTI_PARTY_CALL
SYS_NP_ENABLED
NP_EMT_DFT_PFX
NP_ORIG_B_LOOKUP
NP_LOOKUP_ALL_MSC
CALL_CIRC_ENB
CALL_CIRC_URE_INST
NCF_CHK_RSTRCT
PREFIX_MNP_ENABLED
NCF_LI
SPL_CFWD_RULE
NP_TERM_A_LOOKUP
NP_ORIG_B_LOOKUP
PREFIX_MNP_ENABLED
NUMBER_NORMALIZATION_METHOD
BY_PASS_IVRU_CALL
EMEG_CALL_NUM_NORM_ENABLED
EMERGENCY_CALLS_PASSTHROUGH_DISCARD
FIRST_CALL_INTERCEPT
NTWRK_CALL_FWD_ID
BAL_IN_WELCOME
UNIFIED_CARD_PREFIX
CALLING_CARD_PREFIX
UNIFIED_CARD_PREFIX
SUSP_SUB_INFO_PRMT

OTH_RECH
SPL_CFWD_RULE
CONVERGED_BILLING
MSC_CALL_DUR
CERTIFIED_MSC_CALL_DUR
FREECALL_SERVICECODE
FREECALL_CHARGECODE
NCF_8_BYTES_CRN
BALANCE_PLAY_ALWAYS

Reviewing CallProcessor Process Activity

When the CallProcessor and related process are started, a log file is generated and stored at /home/omni/Logs. This contains the information regarding the process starts and end status.

Expected Input for CallProcessor Process

From IPF:

- ProvideInstructions
- CallInfoFromResource
- ResourceClear
- RoutingInfoFromResource

Expected Output and Location of Output

To IPF:

- CallInfoToResource
- CancelResourceEvent
- ExtendedInfoToResource

List of Success/Failure Criteria for CallProcessor Process

The clear_cause field in CDR record and Usage Record (generated in usage_record_main table) determines the success/failure of the activity.

clear_cause = 16: success,

clear_cause = any other value: fails.

Chapter 26

CAMEL 3 SMS

26

orded (sender
The desti
notifying
ng The noti
ieve The m
t access To

e

CAMEL 3 SMS Functionality Description

This process is responsible for billing of originating SMS services which are built directly into the CAMEL 3 protocol. This process allows/disallows individual SMS transactions for each subscriber. The Comverse ONE solution includes support for billing originating SMS services over CAMEL 3, including support for location-based billing. For location-based billing purposes, CAMEL Phase 3 (CAP3) SMS calls use the LBB_precedence file to determine the locations. The LBB_precedence file supports CAP3 SMS billing by supporting new entries for CAP3 SMS.

Mapping of the SMS call type is done through the Activity Definition table, as is currently done for MO-SMS support via the Event Charging Interface. There is no difference in the entries for CAP3 and those for Event Charging Interface SMS support. If there is no MO-SMS Bearer Application subtype defined for the original call type, the SMS is not allowed. CAP3 SMSs are treated like a call for lifecycle purposes, with the exception of forwarding the call (to the Recharge Server or Customer Center). A subscriber in Fraud Lockout state is not allowed to send SMSs.

Reservations

For one CAMEL 3 SMS activity, there is only one reservation issued by URE_SMS, an SLU service application. When the Subscriber Data Server (SDS) calculates the reservation/consumption for a CAMEL 3 SMS activity, it always uses the first consumption charge of the primary tariff. The reservation is either granted or rejected and the reservation chunk size is always one SMS. There is no such thing as half an SMS or two SMSs since the CAMEL 3 protocol only processes one SMS at a time.

Dependencies on Other Processes

Cap3SMS needs to register to TSP to get static data loaded.

See "[Configuration of CAMEL 3 SMS process](#)" below for instructions.

Running CAMEL 3 SMS Process

To start OMNI, perform these actions:

1. Log in as root, using su password sonora
2. Start OMNI with the /sbin/service omni start command.

To stop OMNI, use this command: /sbin/service omni stop

Configuration of CAMEL 3 SMS process

The detailed steps for the configuration of Cap3Sms are given below:

1. Modify cestart.202 and tnsnames.ora as per each setups requirement. A typical cestart.202 file is shown below:

```
SLF_SMS . 1 DEF DEF 202 100 /home/omni/bin/cap3sms
```

The rest of configuration should be the same as all other applications (corresponding IPF process, IND, needs to be run; SLF_SMS and IND both need to be included in tapdes.202, etc)

2. The following Cap3Sms configuration files are installed as part of the SLF rpm(s) (all in /home/omni/conf):

```
commonCfg.xml
cap3smsCfg.xml
```

tspCfg.xml

Environmental Variables

The required environment variables are created while running configureSLU. You can edit /home/sncpuser/profile/omnisetup.sh to set environmental variables as appropriate.

System Parameters

Set INT_VALUE for the RTNG/SMS_CDR parameter to 1 if RTB style CDR file needs to be generated for a CAP3SMS call. The default value is 0.

Reviewing CAMEL 3 SMS Functionality

When the CAP3SMS and related process are started, a log file is generated and stored at /home/omni/Logs. This contains the information about the process starts and end status.

Below configuration/settings can be verified at any point by using:

```
SLF_SMS>>SLF_SMS,<OPTIONS>;
```

OPTIONS:

HELP

CMPIL-DNMOD-SCRIPTS-PMTS

RESET-MEAS

DUMP-MEAS

DISPLAY-PMTS:ALL

DISPLAY-CFG

CHANGE-PMTS:THRESHOLD-TIMER=NN

CHANGE-PMTS:MSGs-PER-SEC=NN

DUMP-SVC-REC

DUMP-TRANS-BTREE

ENABLE-MODULE=<ALL> or <MODULE_NAME>

DISABLE-MODULE=<ALL> or <MODULE_NAME>

ENABLE-MODULE-MONITORING

DISABLE-MODULE-MONITORING

DISPLAY-MODULE-MONITORING-STATUS

DUMP-MODULE-LIST

DUMP-MODULE-INVOCATION-DATA

DUMP-MODULE-STATS

ENABLE-MODULE-STATS

DISABLE-MODULE-STATS

SUB=<nnn>

DISABLES

Expected Input for CAP3SMS Process

CAP3SMS process receives the request from the IND and processes it. The list of requests the CAP3SMS process receives is given below.

- Provide Instruction (PI)
- Call info from resource (TIFR)
- Resource Clear (RC)

Expected Output and Location of Output

The response that is sent to the IND for the appropriate request is given below:

- Call Info to resource

List of Success/Failure Criteria for CAMEL 3 SMS Functionality

The clear_cause field in CDR record and Usage Record (generated in usage_record_main table) determines the success/failure of the activity.

clear_cause = 16: success,

clear_cause = any other value: fails.

Chapter 27

OR Manager Process

27

orded (sender
The desti
notifying
ng The noti
ieve The m
cT access To

e

OR Manager Process Description

Outage Record Manager (OR Manager) Process collects Outage Records (ORs) for billing events from the service applications and writes ORs to a file. The service applications put one or more billing events (outage records) in a message and then send the message to the OR Manager. OR Manager appends the message to the current opened OR file. When the OR file reaches a specified size or a specified time has elapsed, the OR file is written to the disk. Once the OR file is written to disk the UPA process transfers the OR files to an external host for further processing.

Dependencies on Other Processes

OR Manager depends on service applications (SLF) to send the billing events (outage records) in a timely manner. OR Manager depends on the UPA process to transfer OR files to an external host for further processing in a timely manner.

Running OR Manager Process

OR Manager is enabled by setting the OPT ORM parameter to yes in the sluPlatform file and running configureSLU.

Configuration of OR Manager Process

OR Manager is configured using the following files:

- config
- db.IP.om.\$SHM.pri
- or-filetransfer.cfg.

config

This file is located in \$OMNI_HOME/conf and contains (as an example) the following information:

- RECORD_XCS=
- RECORD_XCS_USER=
- RECORD_XCS_PASSWORD=
- RECORD_XCS_PATH=CDR
- RECORD_SLU_USER=sncpuser
- RECORD_SLU_PASSWORD=sncpuser
- RECORD_NBIP=trm-1
- RECORD_NBIP_USER=bipuser
- RECORD_NBIP_PASSWORD=bipuser
- RECORD_OR=trm-1
- RECORD_OR_USER=oruser
- RECORD_OR_PASSWORD=oruser
- RECORD_CDR_PATH=common/CDR
- RECORD_OR_PATH=common/OR
- RECORD_ORE_PATH=common/ORE
- RECORD_INCLUDE_HEADER=yes
- RECORD_FIELD_DELIMITER=|

- RECORD_HOST_IP=10.230.21.125
- RECORD_XCS_TO_SLU_RESEND_DIRECTORY=/home/omni/kdpm374/resend/XCS
- RECORD_NBIP_TO_SLU_RESEND_DIRECTORY=/home/omni/kdpm374/resend/CDR
- RECORD_XCS_TO_NBIP_RESEND_DIRECTORY=common/resend/CDR

db.IP.om.\$SHM.pri

This distributed DB file contains the following information (minimum information):

CHG-BLLNG:SIZE=10240,PURGE=2,TIME=NO,PERIOD=NO;

or-filetransfer.cfg.

This file, located in /home/omni/conf, is used to define a job to UPA to transfer OR files over SFTP.

Example - or-filetransfer.cfg file

The lines highlighted in bold are the site specific properties that needs to be modified by site personnel. module.id = ALERT_LOG.

```
module.group = log
module.resource.group = host_rg
module.description = transfers OR files to UPM
log.schedule.expression= 0 1/5 * * * ?
file.transfer.total = 1
user.1.name = oruser
user.1.password = oruser
remote.1.host = 10.230.17.100
remote.1.name = /home/oruser/common/OR/IPor.kslu12.*
local.1.name = /data/kslu12/IPor.kslu12.*
user.1.operation = send
delete.1.file = true
```

Reviewing OR Manager Process Activity

OR Manager activity can be verified by checking the creation of OR files with the IPor. <cname>.<seq>.<nnnnnnnnnnnn> file name format in the directory that is defined in the config file (typically /data/<cname>).

Expected Input for OR Manager Process

The expected inputs are the OR records from the service applications.

Expected Output and Location of Output

The expected output is creation of OR files with the IPor. <cname>.<seq>.<nnnnnnnnnnnn> file name format in the directory that is defined in the config file (typically /data/<cname>). OR Manager depends on the UPA to transfer OR files to the external host.

List of Success/Failure Criteria for OR Manager Process

Check if the status has been updated to 4 (complete for Billing first move or Rating side move) or 8 (for Billing second move) in ACCOUNT_MOVE_STATUS.

Troubleshooting Guidelines

Verify if:

- Source and destination directories are correctly defined in the configuration files.
- UPA is correctly configured to periodically transfer OR files to the external host.

Chapter 28

TSP Process - OMNI Mode

orded (sender
The desti
notifying
ng The noti
ieve The m
t access To

TSP Process Description

TSP (the Data Caching Server) reads and caches data from the SDP database in the SLU shared memory and provides interface for applications to retrieve the data.

During startup TSP loads all data for the current major PC version in course of loading the TSP screens sizes of selected tables. This is an effort to detect problems and provide early warning if abnormal table size is detected in the SDP database.

During normal operation, after the new PC version has been successfully propagated, TSP is notified and it loads new data and updates tables in its shared memory segments.

All application processes that require TSP services register with TSP and in turn the TSP notifies them every time the new data is acquired from SDP.

Prerequisites for Running TSP Process

The prerequisites for running TSP process are:

- Rating package has to be installed.
- The configuration files have to be checked or modified
(/home/omni/cfg/commonCfg.xml, /home/omni/cfg/tspCfg.xml,
/home/omni/conf/TspTableSanityList, and so on).

Dependencies on Other Processes

No other special preparations are necessary before running TSP in OMNI mode. There is no direct dependency on other processes.

Running TSP Process

TSP is an instrumented process. It can be started, suspended, resumed and shut down from the Unified Platform and it can send heartbeat messages, alarm information, status information, and statistics to the Unified Platform.

As it is an instrumented process with the PROCESS_SCHEDULE table being used, before starting TSP in UPM, a process-type configuration file with process schedule properties must be defined.

If it is not started by UPA, then it should be started by the user with **root** privileges.

To start and stop TSP in OMNI (on-line) mode, perform the following steps:

3. Log in as **root** using su password **sonora**.
4. Start OMNI using the following command:
`/sbin/service omni start`
5. To stop OMNI, use the following command:
`/sbin/service omni stop`

Configuration of TSP Process

To configure TSP process, perform the following steps:

1. Check and modify cestart.202 and tnsnames.ora as per the setup requirement.
cestart.202 file
OPPS . 1 DEF DEF 202 100 /home/omni/bin/CallProcessor -name OPPS -T

```

TPPS . 1 DEF DEF 202 100 /home/omni/bin/CallProcessor -tpps -name TPPS -T
PMTSVR . 1 DEF DEF 202 101 /home/omni/bin/PmtSvr -d -hbt 180
URE_Q1 . 1 DEF DEF 202 100 /home/omni/bin/ure -Q -ratingdb 1
URE_U1 . 2 DEF DEF 202 100 /home/omni/bin/ure -U -ratingdb 1
URE_Q9 . 1 DEF DEF 202 100 /home/omni/bin/ure -Q -ratingdb 9
URE_U9 . 2 DEF DEF 202 100 /home/omni/bin/ure -U -ratingdb 9
TSP . 1 DEF DEF 202 100 /home/omni/bin/tsp
USSD_SLF . 0 DEF DEF 202 100 /home/omni/bin/ussd_slf

```

2. The following TSP configuration files must be installed as part of the Rating rpm(s) in \$OMNI_HOME/conf and \$OMNI_HOME/profile directories:

- ❑ commonCfg.xml
- ❑ tspCfg.xml
- ❑ TspConfXmptList
- ❑ TspTableSanityList
- ❑ omnisetup.sh

commonCfg.xml

The <DefaultDatabaseInstance> and <DefaultDatabaseType> entries are self-explanatory and the default values are set during product installation. These entries are required by the Security server to retrieve password.

<DefaultOracleUserName> is the user name of the database (SDP) account. The password for the user is retrieved from the Security server by TSP.

<DefaultRatingDbId> is an important parameter. This argument is used to construct the URE process name which communicates with TSP. This process sends Notification message to TSP when new PC version is available in the SDP DB. After successful propagation of PC version the URE_Q<DefaultRatingDbId> process prompts TSP to load new data into shared memory tables.



NOTE

TSP does not require this setting for database (SDP) connection.



NOTE

Database into which TSP connects is determined by DEFAULT_SDP_NODE environment variable. This must correspond to an entry in tnsnames.ora and by default it is not presented in the XML file. However, if the entry is added in the XML file, then it overwrites the value that is set in the environment.

tspCfg.xml

<DefaultTspXmptName> defines the full path name to the list of SDP tables, which should not be loaded into the shared memory by the TSP process.

<db_timeout> defines the maximum time frame in which TSP expects to receive response from SDP.

TspConfXmptList

This is an ASCII file. The file name and the path can be set in tspCfg.xml.

TSP reads this file during startup and also while loading the new propagated PC version. In order to save Shm space and also the loading time, the tables that are not used for online rating can be exempt from loading by TSP. To exempt a table, the table name has to be placed in a file (typed in upper case letters - one name per line).



NOTE

Only tables that are not essential for TSP OMNI operations should be placed in this list.



NOTE

In some instances the TSP table name differs from the SDP table name (in some join tables and so on). Only TSP table names are eligible entries for this file.

TspTableSanityList

This file can be empty, non-existent, or contain entries in the following format:

<table name in capitals>, <min rows loaded>, <deadBand zone in %>

For each table placed in this file, TSP checks:

- If the table loaded into the shared memory contains at least the number of rows specified in the configuration file during startup. This check should alert the operator that the table loaded from the SDP database may not be complete or it is empty.
- The change of a table's size in per-cent units (number of rows) during incremental load. This check helps to recognize unreasonable increase or decrease of a table's size.

Environmental Variables

The environmental variable `DEFAULT_SDP_NODE` is defined in `/home/sncpuser/profile/omnisetup.sh`. This variable can be redefined in any of the TSP configuration XML files mentioned in the ["Configuration of TSP Process"](#).

If `DEFAULT_SDP_NODE` is not defined, then TSP connects to `sdp1_n1` or other SDP instance specified in `tnsnames.ora` file.

Reviewing AMP Process Activity

OMNI debugging commands facilitate important information about overall health of the TSP process.

To start omd tool, connect to SLU and type:

```
omd TSP
```

The `#setm, <hhh>` command sets the trace mask.

To set full mask, type:

```
#setm, x0ffffffff
```

The `#tr` command displays traced events. In course of coming up, TSP logs essential information into the trace buffer, such as, SDP connected, versioned and non-version tables loaded, post-processing, and so on. At the end it logs messages similar to the following lines:

:Connection and load SUCCESSFUL (gTspLoadStatus: 1).

:serverInit() completed.

:Ready to process IPC.

Usually, the last trace lines display communication information indicating application processes registration with TSP.

The `#tsp, version` command displays current PC version loaded into Shm.

The `#tsp, dump-table=<table name>` command displays the content of the Shm table. Content of a table can be saved in a file using shell redirection command.

The `#tsp, dump-table=online_v_list` displays last 5 PC versions loaded into Tsp Shm.

Expected Input for TSP Process

The following configuration files are required for successfully starting the TSP Process:

- `$OMNI_HOME/conf/commonCfg.xml`
- `$OMNI_HOME/conf/tspCfg.xml`
- `$ORACLE_HOME/network/admin/tnsnames.ora`

Expected Output for TSP Process

All TSP important information is saved into a trace buffer.

During startup TSP creates or appends lines to the `$OMNI_HOME/bin/TspDebugInfo.<tspname+$SHM>.log` file. This file contains basic information that can be checked by the development team in case of any major TSP failure.

List of Success/Failure Criteria for AMP Process

Success

- TSP is up and running and *:Ready to process IPC.* is logged in trace.
- Signed up application processes

Failure

- Unable to allocate Shm segments
- Unable to retrieve user password
- Unable to connect to the SDP database
- Unable to size or load a table from SDP
- Failed table sanity check
- Failed table post processing
- Failed to load new PC version data

Troubleshooting Guidelines

Open OMNI debugger and display and analyze the trace buffer using the following commands:

1. `omd TSP`
2. `#setm,x0ffffffff`

3. #tr

Error

TSP cannot allocate SHM segment.

Display TSP trace buffer and check the error message (similar to the following lines):

[14:51:32.308-0516] (ERROR) Error from shmget()

[14:51:32.308-0516] (ERROR) Failed to create shared memory (key: 98255, 680384 bytes) with error 22: Invalid argument

[14:51:32.308-0516] (ERROR) ERROR CREATING Set #2 for NON-VERSION-DATA

TSP is unable to delete already allocated Shm segment or is unable to create a new Shm segment.

Causes

The possible causes are:

- Incorrect user privileges
- Conflict with another TSP process (TSP process already running).

Solution

Correct user privileges. It must be root for TSP in OMNI mode.

Error

TSP is unable to retrieve user password.

Display TSP trace buffer and check the error message that reveals the problem with database password retrieval.

Causes

The possible cause is:

- Incorrect IP address for UPA

Solution

To resolve the issue, use the following steps:

1. Open /etc/hosts file.
2. Check for correct UPA IP address in the UPM section.

Error

TSP is unable to connect to the SDP database.

TSP tries and fails to connect repeatedly.

Causes

The possible causes are:

- Incorrect username/password
- Incorrect SDP entry in the tnsnames.ora file
- Incorrect IP address for sdp?_n? in /etc/hosts file

Solution

Check:

- User name in the `$OMNI_HOME/conf/commonCfg.xml` file.
- SDP entry in `$ORACLE_HOME/network/admin/tnsnames.ora`.

Error

TSP is unable to size or load a table from SDP.

TSP fails to startup with error while trying to calculate the table size or is unable to load data from the SDP table.

Causes

The possible causes are:

- Incorrect or incompatible SDP database
- SDP table missing attribute (incompatible database)

Solution

Check the SDP entry in `$ORACLE_HOME/network/admin/tnsnames.ora`.

Error

Failed Table Sanity Check.

TSP runs TSC after all versioned and non versioned tables have been loaded into a shared memory segment (before they become active).

If tables passes the test, the following message is logged to the TSP trace:

... tables sanity check - PASSED!

If the test for any of the table from the list fails, the details are logged into the TSP trace and TSP startup is aborted. The problem must be investigated and corrected before TSP is successfully brought up again.

For an incremental load it rejects the new PC version and keeps the old PC version tables in Shm. If an incremental load is rejected, then the operator needs to correct the problem and try to reload the rejected PC version manually. Until this is done, TSP continues to work with the older PC version data.

Causes

- SDP table size is larger than the expected (at start-up) size
- SDP table size change is larger than expected (incremental pc v. loading)

Solution

For problem with the startup:

- Check the second argument in `$OMNI_HOME/conf/TspTableSanityList`. **<min rows loaded>**. Check if this number is correct (minimum number of rows to be loaded).
- Modify sanity list file (2nd argument or SDP data).
- Removal of a table entry causes TSP to skip checking this table.



NOTE

Renaming the `TspTableSanityList` file causes TSP to skip TSC completely.

For incremental load fix:

- Check the third argument in `$OMNI_HOME/conf/TspTableSanityList <deadBand zone in %>`. This is the maximum allowable change in table size (increase or decrease) between consecutive PC version loads (in percentage).
- Make appropriate adjustment in file entries.
- Start OMNI debug session using the `omd TSP` command.
- Load updated file into TSP using the `tsp, load-sanitylist` command.
- Start incremental load manually using the `tsp, load-new-pcversion` command.
- If you decide to remove the table from TSC, then:
 - Remove the entry from the `TspTableSanityList` file.
 - Start OMNI debug session using the `omd TSP` command.
 - Load the updated file into TSP using the `tsp, load-sanitylist` command.
 - Start incremental load manually using the `tsp, load-new-pcversion` command.

Error

Failed table post processing

If the table involved in post processing cannot be loaded, then a warning message is recorded in the trace. The table can be compromised even if the TSP continues to load or process the rest of the tables.

Error

Failed to load new PC version data.

Even if prompted TSP does not load data for PC version if it determines that identical PC version has already been loaded into Shm. The message can be found in the trace buffer.

Chapter 29

Weblogic Admin Server - Install and Upgrade Process

Weblogic Admin Server Overview

The Weblogic Admin server enables installing SAPI, RH&T, Async JMS, and OM SAPI using one administration server, which manages all components of the install process rather than having separate administration servers for each of them. An install script is written that installs the Admin server. The existing install scripts are modified for not setting up separate Admin servers.

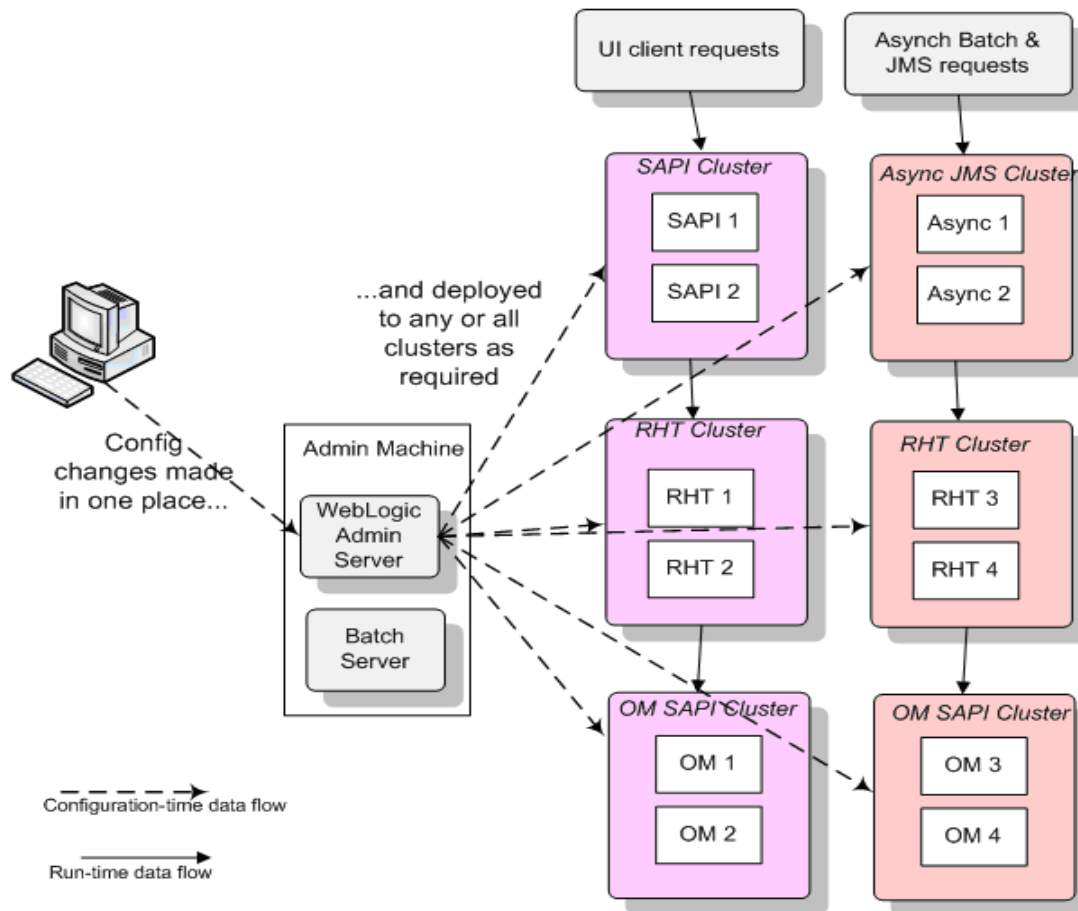
The Weblogic Admin server enables upgrading instead of using Logical Disk/Active Logical Disk (ILD/ALD). Upgrade is an installation process based on upgrade parameters from `configure_XXX.properties`. This reduces downtime significantly during upgrades at least when the database does not need an upgrade. The ILD is used to back up the current install before the installation begins.

Install Process Overview

Earlier each cluster had its own Admin server and it required separate deployments and updates. Using Weblogic Admin server enables only one Admin server to rule all of them and configuration, updates, and installation processes are simplified.

The configuration changes are applied only at one place that is the Weblogic Admin server and the changes are automatically pushed to each managed server.

The Architecture diagram is shown in the figure below.

Figure 47 Architecture

Two streams are defined. One stream is used to handle requests from UIs and the other stream is used to handle requests from asynchronous message processors. In each case, there are several clusters of related servers. Run-time data flow is shown using solid lines; there is no change to the run-time behavior. The configuration information flow is shown using dotted lines. As this diagram shows, however complex the architecture of the site is there is only one administration server.

Upgrade Process Overview

Front End (FE) components upgrade is the process of upgrading a running cluster with a major FE release with minimal down time. During the upgrade of a cluster, each server in the cluster is individually upgraded. Upgrade process includes backup, installing Admin nodes, and restarting servers. The main advantage of upgrading only Admin servers is to reduce down time and has zero down time if multiple managed servers for one cluster are available.

Install Process

Weblogic Admin Server Installation

One Weblogic Administration server is created which controls SAPI, RHT, and ASYNCJMS. A package is installed using the same concepts as all other FE components. The properties file is updated and installation is supported by UFE_Install.sh.

UFE_Install.sh

All FE components are driven by a properties file that contains all required information. For example, directory paths, variable names, Java args, and so on. This property file is used by a common script UFE_Install.sh for installation or upgradation. UFE_Install.sh is packaged into each FE component along with common installation tools.

Admin Install Scripts

The following two files are used for installing Weblogic Administration server:

- install.sh
- install.xml

Installation of Admin server is only required for new installations. Upgrading Admin server is only required when Weblogic software is upgraded.

Admin is a separate package and is installed with rpm, like other FE components. The rpm installer uses UFE_Install.sh as the starting script which in turn calls install.sh. During installation install.sh uses install.xml to create Admin domain. Install.xml uses weblogic's ant task wlserver to create the domain.

Admin is installed as apiuser and apiuser must start and stop server.

Admin Properties

The properties that are required to be set in Configure_admin.properties are:

- INSTALLDIR: Directory in which the install files are stored.
- HOST: Host FQDN.
- HOST_TYPE: Type of box. For example, Linux, AIX, Solaris, and so on.
- JAVA_HOME: Directory path where Java resides.
- BEA_HOME: Top level directory path where Weblogic software is installed.
- WL_HOME: Directory path where Weblogic software is installed.
- CBS_ADMIN_SERVER_INSTALLDIR: Directory in which the Weblogic server is installed.
- ADMIN_DOMAINDIR: Domain directory path.
- ADMIN_DOMAINNAME: Name of the Admin domain.
- ADMIN_WL_HOST: Host where the Admin domain is stored. The default value is HOST.
- ADMIN_WL_PORT: Port of the Admin server.
- ADMIN_WL_PWD: The password of the Weblogic Admin server.
- ADMIN_WL_UID: The username of the Weblogic Admin server.
- ADMIN_WL_USER_MEM_ARGS: Java VM args.

These properties are used by UFE_Install.sh for installation. A package is provided for Admin installation. This package is used for building rpm package for automated installation.

Admin Remove Component Scripts

The scripts are provided to remove following components from Admin server:

- SAPI
- RHT
- SAPI Client
- AsyncJMS

These scripts are located under bin directory where Admin is installed.

Example: /home/admin/server/bin

Admin UPM Scripts

Admin supports the following UPM functions:

- Start server
- Stop server
- Backup server
- Transfer backup file to UPM

The following UPM scripts are provided:

- admin-backup.cfg
- admin.cfg
- admin-filetransfer.cfg

These scripts are placed in \${CBS_ADMIN_SERVER_INSTALLDIR}/config folder.

SAPI Scratch Installation

During scratch install, SAPI is installed on Admin node and all managed nodes. During installation on Admin node, it installs all information for all clusters, datasources, deployment applications, and configuration files. During installation on a managed node, the installation scripts create the domain and add the node to the specified cluster.

SAPI Admin Node Installation

SAPI Properties (When installing on Admin Node)

The installation on Admin node is done once for all clusters.

The SAPI_CLUSTER_INFO property must have unique cluster names and unique multicast address for each cluster. Each pair is separated by a white space and surrounded with quotes.

Example

"cluster1 | 224.0.0.1 cluster2 | 224.0.0.2"



A multicast address is an IP between 224.0.0.0 and 239.255.255.255.

The WORKFLOW_CLUSTER_INFO property must have RHT server:port information for each cluster. Each set is separated by a white space and surrounded with quotes.

Example

```
"CLUSTERNAME | <rhtserver>:<port>,<rhtserver>:<port>
CLUSTERNAME2 | <rhtserver>:<port>,<rhtserver>:<port>"
```

The following additional properties are used while installing SAPI on Admin Node:

- ADMIN_NODE: The values of this property can be Y or N. Set to Y while installing SAPI on Admin node.
- INSTALLDIR: Directory in which the install files are stored.
- HOST: Host FQDN (Fully Qualified Domain Name).
- HOST_TYPE: Type of the box. For example, Linux, AIX, Solaris, HP, and so on.
- JAVA_HOME: Directory path where Java resides.
- BEA_HOME: Top level directory path where Weblogic software is installed.
- WL_HOME: Directory path where Weblogic software is installed.
- SAPI_Databases: Database information for all data sources.

Format for Data Sources

The following WebLogic data source information is available:

- dsname: Unique name for the data source in the WebLogic domain.
- dbhost: Host name or IP address of the machine in which the database resides.
- dbport: Port of the host machine on which the database is communicating.
- dbname: Name of the database.
- dbuserid: User ID of the database.
- dbpwd: Password of the database.
- initcap: Initial number of database connections to establish (should be 0 for a failover backup database)
- maxcap: Maximum number of database connections allowed.
- testonreserve: Test database connections when they are reserved indicator.

0 = false

1 = true

It should be set to 1 (true) for both primary and backup data sources. It is used for a database failover configuration.

- testfreq: Number of seconds to wait between database connection tests for unused connections. In a database failover configuration it controls the frequency of checks on the health of data sources that are marked as unhealthy. When it is set to 0, check is not performed (setting it to 0 is not recommended for a database failover configuration).
- retryfreq: Number of seconds to wait between the attempts to establish connections to the database.
- shrinkfreq: Number of seconds to wait before shrinking a connection pool that has incrementally increased to meet demand.
- trusttime: Number of seconds (since a database connection has been used) to trust that the connection is still viable. Viable connections are not retested during this time period.
- jndinames: Name of the JNDI to access the database.
- faildbhost: Host name or IP address of the machine in which the failover database resides.
- faildbport: Port of the host machine on which the failover database communicates.
- faildbname: Name of the failover database.

- ADMIN_DOMAINDIR: Directory in which Admin domain is located.
- ADMIN_WL_HOST: Host
- ADMIN_WL_PORT: Port for the Admin server. The default value is 7001.
- ADMIN_WL_PWD: The password of the Weblogic Admin server.
- ADMIN_WL_UID: The username of the Weblogic Admin server.
- SECURITY_IP: IP address for the Security server.
- SECURITY_AUDIT_ENABLED: Enables or disables the Security audit. The values are:
 - True
 - False
- SECURITY_AUTHORIZATION_ENABLED: Enables or disables the Security Authorization. The values are:
 - True
 - False

The following additional properties that are used but are always defaulted:

- PRODUCT: Always set to SAPI. This property is used by UFE_Install.sh.
- DELETE_PACKAGE_FILES: The values for this parameter can be Y or N. If it is set to Y, then all package files are deleted. The default value is Y.
- FORCE_DELETE_PACKAGE_FILES: The values for this parameter can be Y or N. If it is set to Y and DELETE_PACKAGE_FILES is set to Y, then installation attempts to remove all package files regardless of whether the files are write-protected or not.
- APPLICATION_SERVER_TYPE: The Application server type. The values can be Weblogic/Websphere.
- SECURITY_PORT: Port for the Security server. The default value is 8443.
- SECURITY_PEER_IP: IP address for the backup Security server. The default value is SECURITY_IP.
- SECURITY_APPLYMASK_ENABLED: The value of this property can be true or false. It is used for ccard.
- SECURITY_APPLYMASK_LEADINGDIGITS: The default value is 0. It is used for ccard.
- SECURITY_APPLYMASK_TRAILINGDIGITS: The default value is 4_xxx. It is used for ccard.
- AUDITING_LOG_ENABLED: The value of this property can be true or false. It is used for auditing.

SAPI Installation Scripts

For installation on Admin node the above properties are set in Configure_sapi.properties file and UFE_Install.sh is executed. When installed with RPM, the RPM install scripts set up the Configure_sapi.properties file and runs UFE_Install.sh in one step. UFE_Install.sh calls components' scripts which know the details on installing the component. UFE_Install.sh is a generic script that is used for all FE components as the starting point and then calls into the lower level scripts which are provided by each component.

The two main files of SAPI are:

- install.sh
This file is used for both creating and updating SAPI component.
- install.xml
This file contains ant task to perform operations that are needed to create and upgrade SAPI components.

These scripts create and configure all files during installation on Admin node and Managed server node.

When installing SAPI on Admin node (ADMIN_NODE = Y), install.sh:

- Creates Cluster. One for each specified in SAPI_CLUSTER_INFO.
- Create Datasources based on SAPI_Databases and are targeted to each cluster.
- Copies configuration properties and xml files into Admin (<admin domain>/config/sapi/<cluster>).
- Deploys SAPI application and targets each cluster.

Admin Node Directory Layout

In addition to the stand Weblogic domain, the following directories are added while installing SAPI on the Admin node.

<admin domain>/config/sapi

- bin
This directory contains all scripts and is linked back to server/bin when installing SAPI on the Managed server. This is required because when a script or any other file is changed it is copied to all managed servers and need not be installed on the managed server.
- config
This directory contains all UPA configuration files.
- <cluster name> - one for each cluster
 - config
It contains configuration properties and xml files specific for a cluster.

SAPI Installation on Managed Server Node

SAPI Properties (When installing on Managed Server Node)

The following properties are used while installing SAPI on the Managed server:

- ADMIN_NODE: The value of this property can be Y or N. Set to N while installing on Managed server node.
- INSTALLDIR: Directory in which Install files are stored.
- HOST: Host FQDN (Fully Qualified Domain Name).
- HOST_TYPE: Type of the box. For example, Linux, AIX, Solaris, HP, and so on.
- JAVA_HOME: Directory path where Java resides.
- BEA_HOME: Top level directory path where Weblogic software is installed.
- WL_HOME: Directory path where Weblogic software is installed.
- CBS_SAPI_SERVER_INSTALLDIR: Directory in which the server is installed.
- SAPI_TURNKEY: The value of the property can be true or false. It is used for setting up environment variables for UPA.
- ADMIN_DOMAINDIR: Directory where the Admin domain is located.
- ADMIN_WL_HOST: Host
- ADMIN_WL_PORT: Port for the Admin server. The default value is 7001.
- ADMIN_WL_PWD: The password of the Weblogic Admin server.
- ADMIN_WL_UID: The username of the Weblogic Admin server.
- SAPI_DOMAINDIR: Directory for SAPI domain.
- SAPI_MANAGED_WEIGHT: Server Weight. The default value is 100.
- SAPI_MANAGED_WL_PORT: Server Port. The default value is 8001.

- SAPI_MANAGED_SERVER_NAME: Name of server. The default value is MS_SAPI_%HOST%.
- SAPI_MANAGED_SERVER_MACHINENAME: Machine name. The default value is %HOST%.
- SAPI_CLUSTER_NAME: Name of cluster for which this server belongs.
- SAPI_WL_PRODUCTION_MODE: The value of this property can be true or false. The default value is true.
- SAPI_WL_JAVA_VM: Java VM Args. The default value is -server.
- SAPI_WL_USER_MEM_ARGS: Java Memory Args.
- enabledEventPublisherNotification: The value of the property can be Y or N. It enables SAPI to publish events.
- EVENTPUBLISHER_NOTIFICATION_CLIENT_JAR_FILES: If enableEventPublisherNotification is set to Y, then this value has jar file for handling events. Full path is required.
- Real time Payment Properties: There are multiple real time payment properties and the descriptions are available in Configure_sapi.properties file.
- SECURITY_IP: IP address for the Security server.
- SECURITY_AUDIT_ENABLED: The value of this property can be true or false.
- SECURITY_AUTHORIZATION_ENABLED: The value of this property can be true or false.
- enabledRCS: The value of this property can be Y or N. This enables SAPI to use RCS.

The following additional properties that are used but are always defaulted:

- PRODUCT: Always set to SAPI. This property is used by UFE_Install.sh.
- DELETE_PACKAGE_FILES: The values for this parameter can be Y or N. If it is set to Y, then all package files are deleted. The default value is Y.
- FORCE_DELETE_PACKAGE_FILES: The values for this parameter can be Y or N. If it is set to Y and DELETE_PACKAGE_FILES is set to Y, then installation attempts to remove all package files regardless of whether the files are write-protected or not.
- APPLICATION_SERVER_TYPE: The Application server type. The value can be Weblogic or Websphere.
- SAPI_LOGDIR: The name of directory where the logs are stored. The root directory is SAPI domain.
- SAPI_LOGNAME: The name of SAPI logs.
- SAPI_DEFAULT_LOG_LEVEL: all < debug < info < warn < error < fatal < off. The default log level is error.
- SECURITY_PORT: Port for the Security server. The default value is 8443.
- SECURITY_PEER_IP: IP address for the backup Security server. The default value is SECURITY_IP.
- SECURITY_APPLYMASK_ENABLED: The values of this property can be true or false. It is used for ccard.
- SECURITY_APPLYMASK_LEADINGDIGITS: The default value is 0. It is used for ccard.
- SECURITY_APPLYMASK_TRAILINGDIGITS: The default value is 4_xxx. It is used for ccard.
- AUDITING_LOG_ENABLED: The value of this property can be true or false. It is used for auditing.

SAPI Installation Scripts

The two main files of SAPI are:

- `install.sh`
This file is used for creating SAPI Managed server.
- `install.xml`
This file contains ant task to perform operations that are needed to create SAPI components.

These scripts create and configure all files while installing SAPI on Managed server node. While installing SAPI on Admin node (`ADMIN_NODE=N`), `install.sh` creates a server and adds it to the cluster specified in `SAPI_CLUSTER_NAME`.

SAPI UPM Scripts

SAPI supports the following UPM functions:

- Start server
- Stop server
- Backup server
- Transfer backup file to UPM
- Reload_cache
- Reload security policy files
- Transfer of audit logs
- Transfer of SAPI logs
- Replacing `log4j.xml` file

The following UPM scripts are provided:

- `reload_cache.cfg`
- `sapiaudit-filetransfer.cfg`
- `sapi-filetransfer.cfg`
- `sapilog4j-replace.cfg`
- `sapilog-filter.cfg`
- `sapi_managed.cfg`
- `reload_policy.cfg`
- `sapi-backup.cfg`
- `sapilog4j-filetransfer.cfg`
- `sapilog-filetransfer.cfg`
- `sapi-restore.cfg`

These scripts are placed in `${CBS_SAPI_SERVER_INSTALLDIR}/config` folder.

SAPI Client Scratch Installation

SAPI client has its own installation and is not part of RHT install. SAPI client must be installed before RHT is installed. SAPI client is only installed on Admin server.

SAPI Client Admin Node Installation

SAPI Client Properties (When installing on Admin Node)

The following properties are used while installing SAPI client on Admin node:

- `PRODUCT`: Always set to SAPI. This property is used by `UFE_Install.sh`.

- UPGRADE: The value of this property can be Y or N. It is set for upgrading the component.
- ADMIN_NODE: It must be set to Y.
- DELETE_PACKAGE_FILES: The value of this parameter can be Y or N. If it is set to Y then all package files are deleted. The default value is Y.
- FORCE_DELETE_PACKAGE_FILES: The value for this parameter can be Y or N. If it is set to Y and DELETE_PACKAGE_FILES is set to Y, then installation attempts to remove all package files regardless of whether the files are write-protected or not.
- INSTALLDIR: Directory in which install files are stored.
- HOST: Host FQDN (Fully Qualified Domain Name).
- HOST_TYPE: Type of the box. For example, Linux, AIX, Solaris, HP, and so on.
- JAVA_HOME: Directory path where Java resides.
- BEA_HOME: Top level directory path where Weblogic software is installed.
- WL_HOME: Directory path where Weblogic software is installed.
- APPLICATION_SERVER_TYPE: The Application server type. The values can be Weblogic/Websphere.
- ADMIN_DOMAINDIR: Directory where the Admin domain is located.
- ADMIN_WL_HOST: Host
- ADMIN_WL_PORT: Port for the Admin server. The default value is 7001.
- ADMIN_WL_PWD: The password of Weblogic Admin server.
- ADMIN_WL_UID: The user name of Weblogic Admin server.
- INSTALL_INTO_PRODUCT=RHT
- WORKFLOW_CLUSTER_INFO: It has RHT cluster names in quotes separated by spaces. Example: "RHT_CLUSTER RHT_CLUSTER_2"

SAPI Client Installation Scripts

The two main files used for installation are:

- install.sh
This file is used for installing SAPI client.
- install.xml
This file contains ant task to perform operations that are needed to create SAPI client component.

RHT Scratch Installation

During scratch install, RHT is installed on the Admin node and all managed nodes. During installation on Admin node, installation installs all information for all clusters, datasources, deployment applications, and configuration files. During installation on a Managed node, the installation scripts create the domain and add the node to the specified cluster.

RHT Admin Node Installation

RHT Properties (When installing on Admin Node)

The installation on Admin node is done once for all clusters.

The WORKFLOW_CLUSTER_INFO parameter must have unique cluster names and unique multicast address for each cluster. Each pair is separated by a white space and surrounded with quotes.

Example

```
"cluster1 | 224.0.0.1 cluster2 | 224.0.0.2"
```



A multicast address is an IP between 224.0.0.0 and 239.255.255.255.

The SAPI_CLUSTER_INFO parameter must have SAPI server:port information for each cluster. Each set is separated by a white space and is surrounded with quotes.

```
"CLUSTERNAME | <sapiserver>:<port>,<sapiserver>:<port>  
CLUSTERNAME2 | <sapiserver>:<port>,<sapiserver>:<port>"
```

The additional properties that are used by RHT installation on Admin node are:

- ADMIN_NODE: The values of this property can be Y or N. Set to Y when installing RHT on Admin node.
- INSTALLDIR: Directory in which the install files are stored.
- HOST: Host FQDN (Fully Qualified Domain Name).
- HOST_TYPE: Type of the box. For example, Linux, AIX, Solaris, HP, and so on.
- JAVA_HOME: Directory path where Java resides.
- BEA_HOME: Top level directory path where Weblogic software is installed.
- WL_HOME: Directory path where Weblogic software is installed.
- WORKFLOW _Databases: Database information for all data sources.

Format for Data Sources

The following Data Source information is available:

- dsname: Unique name for the data source in the Weblogic domain.
- dbhost: Host name or IP address of the machine in which the database resides.
- dbport: Port of the host machine on which the database is communicating.
- dbname: Name of the database.
- dbuserid: User ID of the database.
- dbpwd: Password of the database.
- initcap: Initial number of database connections to establish (should be 0 for a failover backup database).
- maxcap: Maximum number of database connections allowed.
- testonreserve: Test database connections when they are reserved indicator.
0 = false
1 = true

It should be set to 1 (true) for both primary and backup data sources. It is used for a database failover configuration.

- testfreq: Number of seconds to wait between database connection tests for unused connections. In a database failover configuration it controls the frequency of checks on the health of data sources that are marked as unhealthy. When it is set to 0, check is not performed (setting it to 0 is not recommended for a database failover configuration).
- retryfreq: Number of seconds to wait between the attempts to establish connections to the database.
- shrinkfreq: Number of seconds to wait before shrinking a connection pool that has incrementally increased to meet the demand.

- ❑ `trusttime`: Number of seconds (since a database connection has been used) to trust that the connection is still viable. Viable connections are not retested during this time period.
 - ❑ `jndinames`: Name of the JNDI to access the database.
 - ❑ `faildbhost`: Host name or IP address of the machine in which the failover database resides.
 - ❑ `faildbport`: Port of the host machine on which the failover database communicates.
 - ❑ `Faildbname`: Name of the failover database.
- `ADMIN_DOMAINDIR`: Directory in which the Admin domain is located.
- `ADMIN_WL_HOST`: Host
- `ADMIN_WL_PORT`: Port for Admin server. The default value is 7001.
- `ADMIN_WL_PWD`: The password of the Weblogic Admin server.
- `ADMIN_WL_UID`: The username of the Weblogic Admin server.
- `SECURITY_IP`: IP address for the Security server.
- `WORKFLOW_CREATE_SCHEMA`: Defines new database indicator. The default value is N.
- `WORKFLOW_LOAD_MODELS`: Load sample process model indicator. The default value is N.

The following additional properties are used but are always defaulted:

- `PRODUCT`: Always set to SAPI. This property is used by `UFE_Install.sh`.
- `DELETE_PACKAGE_FILES`: The values of this parameter can be Y or N. If it is set to Y, then all package files are deleted. The default value is Y.
- `FORCE_DELETE_PACKAGE_FILES`: The values of this parameter can be Y or N. If it is set to Y and `DELETE_PACKAGE_FILES` is set to Y, then installation attempts to remove all package files regardless of whether files are write-protected or not.
- `APPLICATION_SERVER_TYPE`: The Application server type. The values can be Weblogic or Websphere.
- `SECURITY_PORT`: Port for the Security server. The default value is 8443.
- `SECURITY_PEER_IP`: IP address for the backup Security server. The default value is `SECURITY_IP`.
- `INSTALL_JAR`: `workflow_setup.jar`
- `WORKFLOW_EARFILE`: `wpServerPlus.ear`
- `WORKFLOW_WARFILE`: `workListHandler.war`

RHT Installation Scripts

For installation on Admin node the properties are set in `Configure_rht.properties` file and `UFE_Install.sh` is executed. When installed with rpm, the rpm install scripts set up the `Configure_rht.properties` file and runs `UFE_Install.sh` in one step. `UFE_Install.sh` calls components' scripts which know the details on installing the component. `UFE_Install.sh` is a generic script that is used for all FE components as the starting point and then calls into the lower level scripts which are provided by each component.

The two main files for RHT are:

- `Install_rht.sh`
This file is used for both creating and updating RHT component.
- `Install_rht.xml`
This file contains ant task to perform operations that are needed to create and upgrade RHT components.

These scripts create and configure all files when installing on Admin node and Managed server node.

While installing RHT on Admin node (ADMIN_NODE = Y), install.sh:

- Creates Cluster. One for each specified in WORKFLOW_CLUSTER_INFO.
- Creates Datasources based on WORKFLOW_Databases and is targeted to each cluster.
- Copies configuration properties and xml files into Admin (<admin domain>/config/rht/<cluster>).
- Deploys workflow application and targets each cluster.

RHT Installation on Managed Server Node

RHT Properties (When Installing on Managed Server Node)

The following properties are used while installing RHT on the Managed server:

- ADMIN_NODE: The value of this property can be Y or N. Set to N while installing on Managed server node.
- INSTALLDIR: Directory in which Install files are stored.
- HOST: Host FQDN (Fully Qualified Domain Name).
- HOST_TYPE: Type of the box. For example, Linux, AIX, Solaris, HP, and so on.
- JAVA_HOME: Directory path where Java resides.
- BEA_HOME: Top level directory path where Weblogic software is installed.
- WL_HOME: Directory path where Weblogic software is installed.
- CBS_WORKFLOW_SERVER_INSTALLDIR: Directory in which the server is installed.
- WORKFLOW_TURNKEY: The value of this property can be true or false. It is used for setting up environment variables for UPA.
- ADMIN_DOMAINDIR: Directory where the Admin domain is located.
- ADMIN_WL_HOST: Host
- ADMIN_WL_PORT: Port for the Admin server. The default value is 7001.
- ADMIN_WL_PWD: The password of the Weblogic Admin server.
- ADMIN_WL_UID: The username of the Weblogic Admin server.
- WORKFLOW_DOMAINDIR: Directory for the RHT domain.
- WORKFLOW_MANAGED_WEIGHT: Server Weight. The default value is 100.
- WORKFLOW_MANAGED_WL_PORT: Server Port. The default value is 8001.
- WORKFLOW_MANAGED_SERVER_NAME: Name of the server. The default value is MS_RHT_%HOST%.
- WORKFLOW_MANAGED_SERVER_MACHINENAME: Name of the machine. The default value is %HOST%.
- WORKFLOW_CLUSTER_NAME: Name of the cluster to which this server belongs.
- WORKFLOW_WL_PRODUCTION_MODE: The values of this property can be true or false. The default value is true.
- WORKFLOW_WL_JAVA_VM: Java VM Args. The default value is -server.
- WORKFLOW_WL_USER_MEM_ARGS: Java Memory Args.
- SECURITY_IP: IP for Security server.

The following additional properties are used but are always defaulted:

- PRODUCT: Always set to RHT. This property is used by UFE_Install.sh.
- DELETE_PACKAGE_FILES: The value of this property can be Y or N. If it is set to Y, then all package files are deleted. The default value is Y.

- **FORCE_DELETE_PACKAGE_FILES:** The value of this property can be Y or N. If it is set to Y and **DELETE_PACKAGE_FILES** is set to Y, then installation attempts to remove all package files regardless of whether files are write-protected or not.
- **APPLICATION_SERVER_TYPE:** The value of this property can be Weblogic or Websphere.
- **SECURITY_PORT:** Port for the Security server. The default value is 8443.
- **SECURITY_PEER_IP:** IP for the backup Security server. The default value is **SECURITY_IP**.

AsyncJMS Scratch Installation

During scratch install, AsyncJMS is installed on the Admin node and all Managed nodes. During installation on Admin node, installation installs all information for all clusters, datasources, deployment applications, and configuration files. During installation on a managed node, the installation scripts create the domain and add the node to the specified cluster.

AsyncJMS Admin Node Installation

AsyncJMS Properties (When installing on Admin Node)

Installation on Admin node is done once for all clusters.

The **ASYNCJMS_CLUSTER_INFO** parameter must have unique cluster names and unique multicast address for each cluster. Each pair is separated by a white space and surrounded with quotes.

Example

```
"cluster1 | 224.0.0.1 cluster2 | 224.0.0.2"
```



A multicast address is an IP between 224.0.0.0 and 239.255.255.255.

The **WORKFLOW_CLUSTER_INFO** parameter must have RHT server:port information for each cluster. Each set is separated by a white space and is surrounded with quotes.

Example

```
"CLUSTERNAME | <rhtserver>:<port>,<rhtserver>:<port>  
CLUSTERNAME2 | <rhtserver>:<port>,<rhtserver>:<port>"
```

The following additional properties are used while installing AsyncJMS on the Admin Node:

- **ADMIN_NODE:** The values can be Y or N. Set to Y when installing AsyncJMS on Admin node.
- **INSTALLDIR:** Directory where the install files are stored.
- **HOST:** Host FQDN (Fully Qualified Domain Name).
- **HOST_TYPE:** Type of the box. For example, Linux, AIX, Solaris, HP, and so on.
- **JAVA_HOME:** Directory path where Java resides.
- **BEA_HOME:** Top level directory path where Weblogic software is installed.
- **WL_HOME:** Directory path where Weblogic software is installed.
- **ASYNCJMS_Databases:** Database information for all datasources.

Format for Data Sources

The following WebLogic data source information is available:

- **dsname:** Unique name for the data source in the WebLogic domain.
- **dbhost:** Host name or IP address of the machine in which the database resides.

- dbport: Port of the host machine on which the database communicates.
- dbname: Name of the database.
- dbuserid: User ID of the database.
- dbpwd: Password of the database.
- initcap: Initial number of database connections to establish (should be 0 for a failover backup database).
- maxcap: Maximum number of database connections allowed.
- testonreserve: Test database connections when they are reserved indicator.
0 = false
1 = true
It should be set to 1 (true) for both primary and backup data sources. It is used for a database failover configuration.
- testfreq: Number of seconds to wait between database connection tests for unused connections. In a database failover configuration it controls the frequency of checks on the health of data sources that are marked as unhealthy. When it is set to 0, check is not performed (setting it to 0 is not recommended for a database failover configuration).
- retryfreq: Number of seconds to wait between the attempts to establish connections to the database.
- shrinkfreq: Number of seconds to wait before shrinking a connection pool that has incrementally increased to meet the demand.
- trusttime: Number of seconds (since a database connection has been used) to trust that the connection is still viable. Viable connections are not retested during this time period.
- jndinames: Name of the JNDI to access the database.
- faildbhost: Host name or IP address of the machine in which the failover database resides.
- faildbport: Port of the host machine on which the failover database is communicating.
- Faildbname: Name of the failover database.



The WebLogic data source information must be specified for each WebLogic data source in the sequence as mentioned above separated by blanks. Separate the configuration data for multiple data sources with a semicolon (;).

- ADMIN_DOMAINDIR: Directory in which the Admin domain is located.
- ADMIN_WL_HOST: Host
- ADMIN_WL_PORT: Port for the Admin server. The default value is 7001.
- ADMIN_WL_PWD: The password of the Weblogic Admin server.
- ADMIN_WL_UID: The username of the Weblogic Admin server.
- SECURITY_IP: IP for the Security server.
- SECURITY_AUDIT_ENABLED: The value of this property can be true or false.
- SECURITY_AUTHORIZATION_ENABLED: The value of this property can be true or false.

The following additional properties are used but are always defaulted:

- **PRODUCT:** Always set to ASYNCJMS. This property is used by UFE_Install.sh.
- **DELETE_PACKAGE_FILES:** The value of this property can be Y or N. If it is set to Y, then all package files are deleted. The default value is Y.
- **FORCE_DELETE_PACKAGE_FILES:** The value of this property can be Y or N. if it is set to Y and DELETE_PACKAGE_FILES is set to Y, then installation attempts to remove all package files regardless of whether the files are write-protected or not.
- **APPLICATION_SERVER_TYPE:** The value of this property can be Weblogic or Websphere.
- **SECURITY_PORT:** Port for the Security server. The default value is 8443.
- **SECURITY_PEER_IP:** IP for backup Security server. The default value is SECURITY_IP.
- **SECURITY_APPLYMASK_ENABLED:** The value of this property can be true or false. It is used for ccard.
- **SECURITY_APPLYMASK_LEADINGDIGITS:** The default value is 0. It is used for ccard.
- **SECURITY_APPLYMASK_TRAILINGDIGITS:** The default value is 4_xxx. It is used for ccard.
- **AUDITING_LOG_ENABLED:** The value of this property can be true or false. It is used for auditing.

AsyncJMS Installation Scripts

For installation on Admin node, the properties are set in Configure_asyncjms.properties file and UFE_Install.sh is executed. When installed with rpm, the rpm install scripts set up the Configure_asyncjms.properties file and run UFE_Install.sh in one step. UFE_Install.sh calls components' scripts which know the details on installing the component. UFE_Install.sh is a generic script that is used for all FE components as the starting point and then calls into the lower level scripts which are provided by each component.

The two main files for AsyncJMS are:

- **install.sh**
This file is used for both creating and updating AsyncJMS component.
- **install.xml**
This file contains ant task to perform operations that are needed to create and upgrade AsyncJMS components.

These scripts create and configure all files while installing on Admin node and Managed server node.

When installing on Admin node (ADMIN_NODE=Y), install.sh:

- Creates Cluster. One for each specified in ASYNCJMS_CLUSTER_INFO.
- Creates Datasources based on ASYNCJMS_Databases and is targeted to each cluster.
- Copies configuration properties and xml files into admin (<admin domain>/config/asyncjms/<cluster>)
- Deploys asyncjms application and targets each cluster.

Admin Node Directory Layout

In addition to the stand Weblogic domain the following directories are added while installing AsyncJMS on Admin node.

<admin domain>/config/asyncjms

- **bin**
This directory contains all scripts and is linked back to server/bin while installing AsyncJMS on the Managed server. This is needed because when a script or any other file is

changed it is copied to all Managed servers and need not be installed on the Managed server.

- config
This directory contains all the UPA configuration files.
- <cluster name> - one for each cluster
 - config
This directory contains configuration properties and xml files specific for a cluster.

AsyncJMS Installation on Managed Server Node

ASYNCJMS Properties (When installing on Managed Server Node)

The following properties are used while installing AsyncJMS on the Managed server:

- ADMIN_NODE: The value of this property can be Y or N. It is set to N when installing on Managed server node.
- INSTALLDIR: Directory where the install files are stored.
- HOST: Host FQDN (Fully Qualified Domain Name).
- HOST_TYPE: Type of the box. For example, Linux, AIX, Solaris, HP and so on.
- JAVA_HOME: Directory path where Java resides.
- BEA_HOME: Top level directory path where Weblogic software is installed.
- WL_HOME: Directory path where Weblogic software is installed.
- CBS_ASYNCJMS_SERVER_INSTALLDIR: Directory where server is installed.
- ASYNCJMS_TURNKEY: The value of this property can be true or false. It is used for setting up environment variables for UPA.
- ADMIN_DOMAINDIR: Directory where the Admin domain is located.
- ADMIN_WL_HOST: Host
- ADMIN_WL_PORT: Port for Admin server. The default value is 7001.
- ADMIN_WL_PWD: The password of the Weblogic Admin server.
- ADMIN_WL_UID: The username of the Weblogic Admin server.
- ASYNCJMS_DOMAINDIR: Directory for AsyncJMS domain.
- ASYNCJMS_MANAGED_WEIGHT: Server Weight. The default value is 100.
- ASYNCJMS_MANAGED_WL_PORT: Server Port. The default value is 8001.
- ASYNCJMS_MANAGED_SERVER_NAME: Name of the server. The default value is MS_ASYNCJMS_%HOST%.
- ASYNCJMS_MANAGED_SERVER_MACHINENAME: Name of the Machine. The default value is %HOST%.
- ASYNCJMS_CLUSTER_NAME: Name of the cluster to which the server belongs.
- ASYNCJMS_WL_PRODUCTION_MODE: The value of the property can be true or false. The default value is true.
- ASYNCJMS_WL_JAVA_VM: Java VM Args. The default value is -server.
- ASYNCJMS_WL_USER_MEM_ARGS: Java Memory Args.
- enabledEventPublisherNotification: The value of this property can be Y or N. This enables AsyncJMS to publish events.
- EVENTPUBLISHER_NOTIFICATION_CLIENT_JAR_FILES: If enableEventPublisherNotification is set to Y, then this value has jar file for handling events. The complete path is required.
- Real time payment properties: There are multiple real time payment properties and the descriptions are covered in Configure_asyncjms.properties file.

- SECURITY_IP: IP for the Security server.
- SECURITY_AUDIT_ENABLED: The value of this property can be true or false.
- SECURITY_AUTHORIZATION_ENABLED: The value of this property can be true or false.
- enabledRCS: The value of this property can be true or false. It enables AsyncJMS to use RCS.

The following additional properties are used but are always defaulted:

- PRODUCT: Always set to ASYNCJMS. This property is used by UFE_Install.sh.
- DELETE_PACKAGE_FILES: The value of this property can be Y or N. If it is set to Y, then all package files are deleted. The default value is Y.
- FORCE_DELETE_PACKAGE_FILES: The value of this property is Y or N. If it is set to Y and DELETE_PACKAGE_FILES is set to Y, then installation attempts to remove all package files regardless of whether the files are write-protected or not.
- APPLICATION_SERVER_TYPE: The value of this property can be Weblogic or Websphere.
- ASYNCJMS_LOGDIR: Name of the directory in which the logs are stored. Root directory is AsyncJMS domain.
- ASYNCJMS_LOGNAME: Name for AsyncJMS logs.
- ASYNCJMS_DEFAULT_LOG_LEVEL: all < debug < info < warn < error < fatal < off. The default value is error.
- SECURITY_PORT: Port of the Security server. The default value is 8443.
- SECURITY_PEER_IP: IP for backup Security server. The default value is SECURITY_IP.
- SECURITY_APPLYMASK_ENABLED: The value can be true or false. It is used for ccard.
- SECURITY_APPLYMASK_LEADINGDIGITS: The default value is 0. It is used for ccard.
- SECURITY_APPLYMASK_TRAILINGDIGITS: The default value is 4_xxx. It is used for ccard.
- AUDITING_LOG_ENABLED: The value can be true or false. It is used for auditing.

ASYNCJMS Installation Scripts

The two main files for AsyncJMS installation are:

- install.sh
This file is used for creating AsyncJMS Managed server.
- install.xml
This file contains ant task to perform operations that are needed to create AsyncJMS components.

These scripts create and configure all files when installing on Managed server node.

When installing on Managed node (ADMIN_NODE=N), install.sh creates the server and adds it to the cluster specified in ASYNCJMS_CLUSTER_NAME.

ASYNCJMS UPM Scripts

API supports the following UPM functions:

- Start server
- Stop server
- Backup server
- Transfer backup file to UPM
- Reload_cache
- Reload security policy files
- Transfer of audit logs
- Transfer of asyncjms logs

- Replacing log4j.xml file

The following UPM scripts are provided:

- reload_cache.cfg
- asyncjmsaudit-filetransfer.cfg
- asyncjms-filetransfer.cfg
- asyncjmslog4j-replace.cfg
- asyncjmslog-filter.cfg
- asyncjms_managed.cfg
- reload_policy.cfg
- asyncjms-backup.cfg
- asyncjmslog4j-filetransfer.cfg
- asyncjmslog-filetransfer.cfg
- asyncjms-restore.cfg

These scripts are placed in `${CBS_ASYNCJMS_SERVER_INSTALLDIR}/config`.

Upgrade Process

The upgrade of front end components is an installation process. Back up the active Admin server and then install Admin nodes on the Admin server.

Normal installation and upgrade installation can be distinguished by configuring the following three parameters in `Configure_XXX.properties`. Based on the upgrade requirements the following parameters have to be set:

- `UPGRADE = Y`
- `UPGRADE_EAR = Y`
- `UPGRADE_DATASOURCES = Y`

To upgrade an EAR, set the following parameters to Y.

- `UPGRADE = Y`
- `UPGRADE_EAR = Y`

To upgrade Datasources, set the following parameters to Y.

- `UPGRADE = Y`
- `UPGRADE_DATASOURCES = Y`

Back Up Admin Server

Backing up Admin server is mandatory step before upgrade installation.

Verify if the Admin server is running or not. If it is running, then stop the Admin server and use file system copy to take backup.

Stop Admin Server

1. `su - apiuser`
2. `cd /home/admin/server/bin`
3. `./stop_admin.sh`

File System Backup

The file system backup is stored in `cp /home/admin /home/admin_backup`.

Upgrade SAPI Admin Node

SAPI upgrade installation on Admin node is done once for all clusters. The properties that are used by SAPI upgrade on Admin Server are:

- `UPGRADE = Y`
- `UPGRADE_EAR = Y`
- `UPGRADE_DATASOURCES = N`
- `ADMIN_NODE = Y`: Set to Y while installing SAPI on Admin node.
- `INSTALLDIR = /home/sapi` (Directory in which install files are stored).
- `HOST`: Host FQDN (Fully Qualified Domain Name).
- `HOST_TYPE`: Type of the box. For example, Linux, AIX, Solaris, HP, and so on.
- `JAVA_HOME`: Directory path where Java resides.
- `BEA_HOME`: Top level directory path where Weblogic software is installed.
- `WL_HOME`: Directory path where Weblogic software is installed.
- `SAPI_Databases`: This parameter is only configured if `UPGRADE_DATASOURCES = Y`.

Upgrade SAPI Client

SAPI client upgrade installation on Admin node is done once for all clusters. The properties that are used by RHT upgrade on Admin server are:

- `UPGRADE = Y`
- `ADMIN_NODE = Y`: Set to Y while installing SAPI on Admin node.
- `INSTALLDIR = /home/rht` (Directory in which install files are stored).
- `HOST`: Host FQDN (Fully Qualified Domain Name).
- `HOST_TYPE`: Type of the box. For example, Linux, AIX, Solaris, HP, and so on.
- `JAVA_HOME`: Directory path where Java resides
- `BEA_HOME`: Top level directory path where Weblogic software is installed.
- `WL_HOME`: Directory path where Weblogic software is installed.

Upgrade RHT Admin Node

RHT upgrade installation on Admin node is done once for all clusters. The properties that are used by RHT upgrade on Admin server are:

- `UPGRADE = Y`
- `UPGRADE_EAR = Y`
- `UPGRADE_DATASOURCES = N`
- `ADMIN_NODE = Y`: Set to Y while installing RHT on Admin node.
- `INSTALLDIR = /home/rht` (Directory where install files are stored).
- `HOST`: Host FQDN (Fully Qualified Domain Name).
- `HOST_TYPE`: Type of the box. For example, Linux, AIX, Solaris, HP, and so on.
- `JAVA_HOME`: Directory path where Java resides.
- `BEA_HOME`: Top level directory path where Weblogic software is installed.
- `WL_HOME`: Directory path where Weblogic software is installed.

- WORKFLOW_Databases: This parameter is only configured if UPGRADE_DATASOURCES = Y.

Upgrade AsyncJMS Admin Node

AsyncJMS upgrade installation on Admin node is done once for all clusters. The properties that are used by AsyncJMS upgrade on Admin server are:

- UPGRADE = Y
- UPGRADE_EAR = Y
- ADMIN_NODE = Y: Set to Y while installing AsyncJMS on Admin node.
- INSTALLDIR: /home/asyncjms (Directory in which install files are stored.)
- HOST: Host FQDN (Fully Qualified Domain Name.)
- HOST_TYPE: Type of the box. For example, Linux, AIX, Solaris, HP, and so on.
- JAVA_HOME: Directory path where Java resides.
- BEA_HOME: Top level directory path where Weblogic software is installed.
- WL_HOME: Directory path where Weblogic software is installed.
- ASYNCJMS_Databases: This parameter is only configured if UPGRADE_DATASOURCES = Y.

Restarting Servers

Restarting Admin server after upgrading Admin server is not required. Restarting Admin server is only required in case of Datasource upgrade, otherwise it is not required.

Restarting Managed servers is mandatory. Admin server configuration changes are updated in Managed Server during restart of the Managed servers.

Add Remove Scripts

Adding SAPI Managed Server to Existing Cluster

To add new SAPI Managed server to the existing cluster, perform the following steps:

1. Install Managed server by updating configure_sapi.properties.
2. Execute UFE_Install.sh.
3. After successful installation, run addSAPINode.sh to update CCBSClientConfiguration.xml.

Example

```
cd /home/admin/server/bin
./addSAPINode.sh RHT_CLUSTER_CSM <HOST> <PORT>
```

where,

- RHT_CLUSTER_CSM: This is the SAPI cluster that is updated with the new RHT server.
- HOST: The host name of the SAPI server.
- PORT: The port number of the SAPI server.

Adding RHT Managed Server to Existing Cluster

To add new RHT Managed server to the existing cluster, perform the following steps:

1. Install Managed server by updating `configure_rht.properties`.
2. Execute `UFE_Install.sh`.
3. After successful installation, run `addRHTNode.sh` to update `Workpoint-client.properties`.

Example

```
cd /home/admin/server/bin
./addRHTNode.sh SAPI_CLUSTER_CSM <HOST> <PORT>
```

where,

- **SAPI_CLUSTER_CSM:** It is the RHT cluster which has to be updated with the new SAPI server.
- **HOST:** The host name of the RHT server.
- **PORT:** The port number of the RHT server.

Remove SAPI Managed Server from Existing Cluster

To remove SAPI Managed server from the existing cluster, run `removeSAPINode.sh` script on Admin server. It deletes the SAPI Managed server and updates `CCBSClientConfiguration.xml`.

Example

```
cd /home/admin/server/bin
./removeSAPINode.sh <MANAGED_SERVER_NAME> <HOST> <PORT>
```

where,

- **MANAGED_SERVER_NAME:** The name of the SAPI Managed server.
- **HOST:** The host name of the SAPI server.
- **PORT:** The port number of the SAPI server.

Remove RHT Managed Server from Existing Cluster

To remove RHT Managed server from existing cluster, run `removeRHTNode.sh` script on the Admin server. It deletes the RHT Managed server and updates the entries in `Workpoint-client.properties`.

Example

```
cd /home/admin/server/bin
./removeRHTNode.sh <MANAGED_SERVER_NAME> <HOST> <PORT>
```

where,

- **MANAGED_SERVER_NAME:** The Managed sever name of the Workflow.
- **RHT1:** RHT server HOST name
- **PORT:** RHT server PORT

Adding New Cluster for Component

Adding SAPI Cluster

To create a new SAPI cluster on Admin node, perform the following steps:

1. Set the new cluster information in the Configure_sapi.properties file.
2. Run addSAPICluster.sh.

install_sapi.xml

It contains ant task to perform operations that are needed to create SAPI Cluster component.

Adding RHT Cluster

To create a new RHT cluster on Admin node, perform the following steps:

1. Set the new cluster information in the Configure_rht.properties file.
2. Run addRHTCluster.sh.

install_rht.xml

It contains ant task to perform operations that are needed to create RHT Cluster component.

Adding ASYNCJMS Cluster

To create a new ASYNCJMS cluster on Admin node, perform the following steps:

1. Set the new cluster information in the Configure_asyncjms.properties file.
2. Run addJMSCluster.sh.

install_asyncjms.xml

It contains ant task to perform operations that are needed to create ASYNCJMS Cluster component.

Remove Cluster for Component

To remove cluster for a component, run remove_XXX_Cluster.sh by passing parameter of the cluster name on the Admin server. It removes the cluster.

Example

```
cd /home/admin/server/bin
./remove_sapi_cluster.sh SAPI_CLUSTER -> Removes SAPI Cluster
./remove_rht_cluster.sh RHT_CLUSTER -> Removes RHT Cluster
./remove_jms_cluster.sh JMS_CLUSTER -> Removes the JMS Cluster
```


Index

A

ASU 1, 3

C

Configuration 3, 5, 3

CTI 1

custDB 3, 4

E

Environmental 4

Expected 8, 4, 9, 10

Extract 7

H

HOST 8

L

List 8, 4, 10, 6

O

Operational 1

Optimal 4, 9, 6

ORD 3

P

Prerequisites 3

R

Reviewing 8, 4, 6, 5

RH 1, 3

S

SAPI 1, 3

T

Troubleshooting 6

TSS 1, 3

U

UFA 1, 3

UFP 3

URC 1, 3

URE 1, 3

URR 8

Use 4

