amdocs billingplatform

Amdocs Billing Platform 6.0

Upgrade Guide (CDE)

Document Information

Software Version:	**6.0**
Publication Date:	**January 2005; Updated August 2005 for SP4**
Catalog Number:	**212290**

Contents

1. INTRODUCTION

Amdocs Billing Platform provides telecommunication companies with an easily implemented Customer Care & Billing system.

Amdocs Billing Platform is comprised of software packages that run on a distributed, client-server UNIX platform. The front-end client for customer services is Web-based, while the business parameter clients are MS Windows based. The different Amdocs components work together with a variety of third-party software and in-house tools developed by Amdocs.

The upgrade of Amdocs Billing Platform (ABP) from version 5.5 to version 6.0 requires the configuration, synchronization, and upgrade of these various and interdependent hardware and software packages.

The Amdocs Product Development Organization (APDO) team has produced a comprehensive *Amdocs Billing 5.5 to 6.0 Upgrade Kit* which provides a set of documents with step-by-step instructions, scripts and files to enable each Amdocs account to upgrade the version 5.5 core layer and data to version 6.0. Any additional changes made by an account's Infrastructure team should be packaged and added to this Upgrade Kit and used to upgrade the customization areas before performing the upgrade.

Version 6.0 is the first sourceless version of the ABP product and hence, from this version, the upgrade process will be separated into two different areas: the Customization Development Environment (CDE) area and the runtime (RT) area. For each area, a different document is describing the step by step upgrade procedure.

The following document describes the Customization Development Environment area step by step upgrade.

Upgrade Overview

The Amdocs Billing Platform upgrade can be divided into the following main areas, in accordance with the skill set required of the installer:

Infrastructure Upgrade Procedure

This part of the upgrade is performed by the account's Infrastructure team that supports the CDE, testing and production teams.

The Infrastructure Upgrade Procedure provides step-by-step instructions that enable a smooth upgrade to the Amdocs Billing Platform version 6.0 environments. It provides a complete list of the prerequisite software, modified hardware requirements and recommended settings for important system parameters.

The Infrastructure Upgrade Procedure is provided as part of this document (in Chapter 3) and is essential for upgrading the Production environment.

Development Upgrade Procedure

This part of the upgrade is performed by the client or Amdocs development team that is responsible for specific customizations and their implementation layer. This stage also involves activities performed by the infrastructure team to prepare the CDE environment that is a prerequisite for the development team.

The Development Upgrade Procedure provides step-by-step instructions for each application area, which enable a smooth upgrade to the version CDE environment. This enables the development of new customization and implementation functionality in the customization and implementation layers, while maintaining existing functionality intact in the core layer.

The Development Upgrade Procedure is provided as part of this document (in Chapter 4).

As some additional activities may be performed by each account during the development phase, the account's Infrastructure team should create the programs to upgrade the customization areas, while the Amdocs Product Development Organization (APDO) team supplies the core upgrade. These additional programs should be added to the Upgrade Kit and activated together with the core upgrade when running the testing environment upgrade.

Database Upgrade Procedure

This part of the upgrade is performed by the client or Amdocs Infrastructure team. It involves upgrading the Tiger tool, adding customization to the Oracle and TimesTen databases and performing post upgrade tasks.

Client upgrade is performed in parallel.

Scope of This Document

This document provides general upgrade guidelines, as well as step-by-step instructions on how to upgrade Amdocs Billing Platform CDE from version 5.5 to version 6.0.

See Chapter 2 for an overview of the main stages of the upgrade process.

Target Audience

This document is intended for all personnel that are involved in the planning, testing and implementation of the upgrade process. This document is targeted mainly at the account's CDE, Infrastructure and Testing personnel that will actually carry out the upgrade procedures. It is also relevant for IT managers and personnel that want to become familiar with the upgrade process and guidelines.

Prior Knowledge

This Upgrade Guide assumes the user is familiar with Amdocs Billing Platform, its UNIX and Windows® components, Oracle databases, standard system administration practices, and basic third-party knowledge.

Assumptions

The following assumptions apply to the *Amdocs Billing 5.5 to 6.0 Upgrade Kit*:

- The upgrade applies to the conversion of Amdocs Billing Platform version 5.5 to version 6.0.
- The Upgrade Kit covers only the core modules supplied in Amdocs Billing Platform version 6.0.
- As the upgrade process is not recoverable, a rollback to the initial state should be made using relevant backups, when required.
- Client packages are reinstalled using the supplied installation packages and instructions.
- The upgrade guidelines supplied in this document are those recommended by Amdocs product development organization. Different upgrade strategies can be planned by an account according to the specific requirements of the customer.
- The developers are familiar with the Release Notes documents supplied as part of the version 6.0 document deliveries, and the database changes described therein.
- The customer has installed the relevant version 5.5 service pack (currently Service Pack 10).
- Oracle software exists on all servers.
- TimesTen software exists on all servers (if applicable).

The TimesTen database is optional, depending on the site architecture.

Related Documents

The following documents complete the Upgrade Guide document and are referred to explicitly in the above document, when required:

- ABP Release Notes for each application
- ABP Infrastructure Release Notes
- ABP Server Pre-Installation Guide, including:
  - Installation Overview
  - Server Hardware and Software Requirements
  - UNIX Setting & Configuration Guide
  - Database Setting & Configuration Guide
- ABP Server Installation Guide (CDE)

- ABP Server Installation Guide (Runtime)
- ABP Client Installation Guide (CDE)
- ABP Upgrade Guide (Runtime)

Concepts and Terms

The following terms and acronyms are used in this document:

Term	Description
CC	CC (for Configuration Control) is a tool, used for coordinating the software versions between those on site and those of the Amdocs Development Center.
CDE	Customization Development Environment is a framework for further development of ABP product according to customer's requirements
LEL	Local Extension Layer – Equivalent to a customization of Amdocs Billing Platform
Server	When used in this document, the term the server, refers to one of the following, depending on the context: CC server UNIX machine
Tiger	Tiger is a tool for the creation and maintenance of databases. It comes in two versions: a stand-alone version and a version used within AIM.
ADBA Repository	The Application Database Administration (ADBA) repository is the database account that contains the information used by Tiger to create database user accounts and their content.
Xtra-C	Xtra-C is an improved Software Configuration Management (SCM) tool. It is based on CCC/Harvest. Xtra-C delivers a single solution for tracking software changes and managing the application development process in distributed environments.

2. CDE UPGRADE OVERVIEW

This chapter describes general guidelines and information regarding the upgrade from development environment in version 5.5 to development environment in version 6.0 called CDE – Customization Development Environment.

It also describes the detailed upgrade workflow.

General Upgrade Workflow

The general installation workflow consists of five stages:

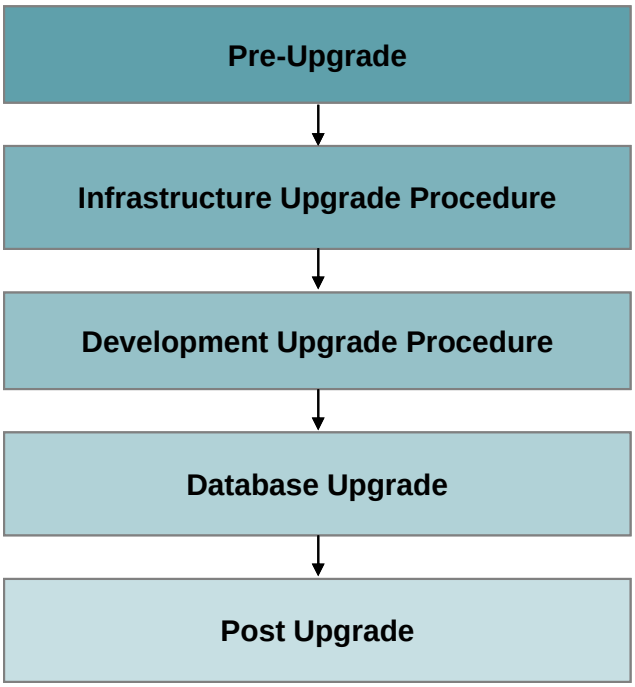


Figure 2-1: General Upgrade Workflow

Pre Upgrade

Preparation for the upgrade process should start approximately two weeks before the upgrade date and involves performing general activities, such as verifying all third-party software versions and licenses, and that all the required patches were loaded on the version 5.5 system.

It is recommended to include as many activities as possible in the preparation stage, and not in the pre-upgrade and upgrade stages, to ensure that the duration of the upgrade process itself is as short as possible.

It is important to complete all relevant preparations at least two days before the actual upgrade date.

Prior to starting the upgrade process, various activities need to be performed such as running successful CC build on version 5.5 to make sure that the sources we will copy to version 6.0 are ready for CC build. After that, a final load to the Tiger repository should be done in order to make sure the Tiger repository is updated with version 5.5 CC. Additional activities are: backing up, running end of day processes, pausing reference data updates, shutting down processes.

The pre-upgrade stage is covered by the Amdocs Billing Platform Server Pre-Installation Guide.

Infrastructure Upgrade

Migrating to next core version consists of the following steps:

- Installing/upgrading third party software
- Upgrading Xtra-C
- Installing SDK version 6.0
- Adjusting the LEL layer to achieve a full build
- Creating an upgrade kit for LEL on top of the core kit

The infrastructure upgrade stage is covered in Chapter 3 of this document.

Development Upgrade

The upgrade of Amdocs Billing Platform (ABP) from version 5.5 to version 6.0 involves several activities that are performed by the CDE team and infrastructure team concurrently.

Specific changes need to be made by the customization developers in different CDE areas, for each application. Typical changes are, for example:

- Make file adjustments
- Data types changes
- Changed/deleted API
- Changed/deleted/new Exit Points
- Application related issues
- Deprecated API's

The output of this step is an upgrade kit that is ready for testing by the system test team. This kit includes all the infrastructure upgrade programs, the CDE customization areas and valid storage for running the tests.

The infrastructure upgrade stage is covered in Chapter 4 of this document.

Database Upgrade

The database upgrade consists of the following steps:

- Upgrading the Tiger tool and repository
- Preparing Oracle and TimesTen databases upgrade using the Tiger repository

Post-Upgrade Activities

Following the completion of the upgrade process, some post-upgrade activities are performed, such as synchronizing reference tables, removing redundant applications, backing up the database, performing sanity checks.

The database upgrade and post-upgrade stages are covered in Chapter 5 of this document.

For client installation information, refer to the Client Installation guide (CDE).

CDE Environment Structure

In version 5.5, the account got in the upgrade kit export of sources from APDO CC, imported them into his private CC, upgraded the customization sources, adapted the gaps and then ran the CC build. After successful build of the core and the LEL (local extension layer) the ADBA loaded the generated gdd files into Tiger and created the DB schema. This schema included both the core and the LEL extensions.

In version 6.0, the product moved into the sourceless world. In this new world, the CDE environment looks different. The upgrade kit includes an SDK (Software Development Kit) instead of getting the sources. The SDK is a pack of directories that include header files and shared libraries that are used for the customization. The SDK does not include executables.

In v6.0 we called the development environment CDE – Customization Development Environment.

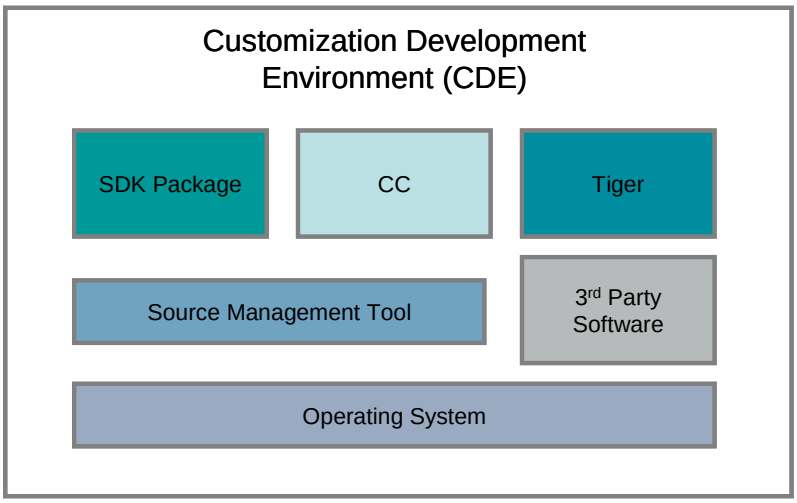


Figure 2-2: CDE Structure

As you can see in the CDE structure, it includes a few parts: SDK package and Configuration control (CC).

SDK Package

An SDK (Software Development Kit) installation package that includes the following elements of ABP version 6.0:

- Shared libraries
- Header files
- EJB & classes jars
- Java, C++ data type and XGEN generators
- Metadata (Fox configuration files, Publish/Subscribe XSD files, Sonar templates)
- DB objects
- CDE configuration files
- Documentation
- Code samples
- CC Manager tools – the SDK already includes the ccmngr tools.
- Make files.

The following figure presents the SDK version 6.0 structures.

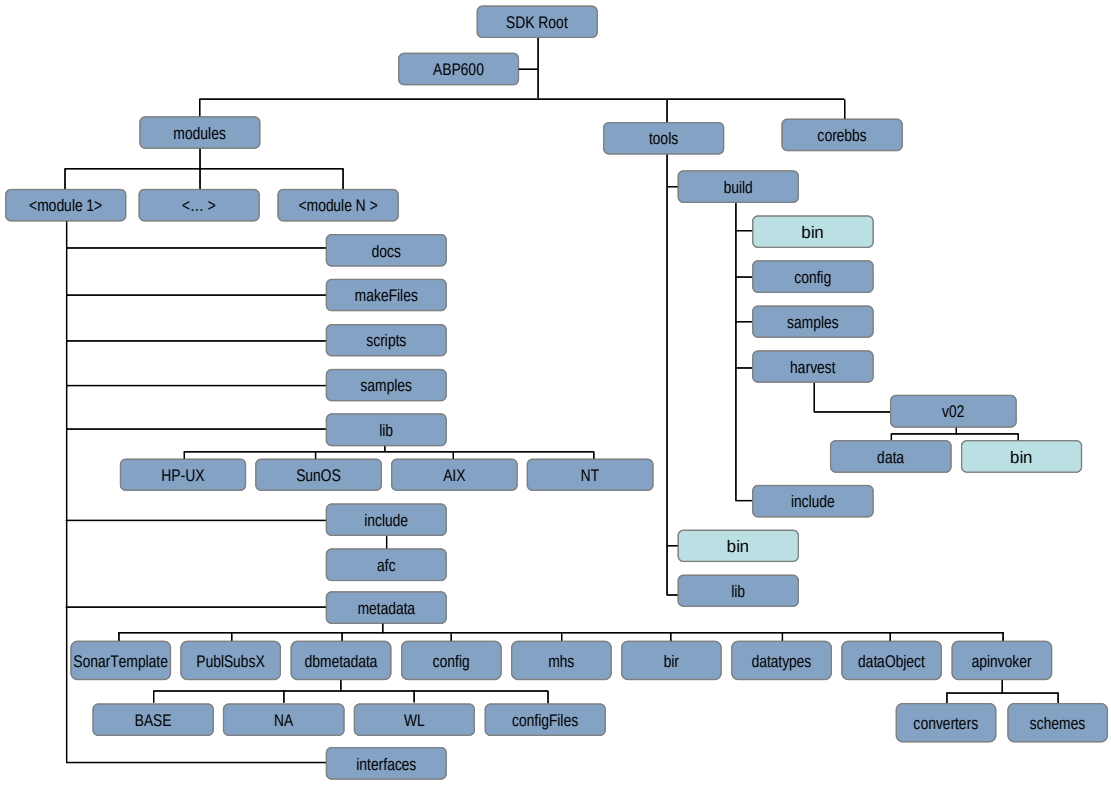


Figure 2-3: SDK Package Version 6.0 Structure

CC and Source Management Tools

These elements contain the following:

1. Source Management Tool – Manages the source code, including the control, auditing and access of source files.
2. Configuration Management – Defines the components of a project, the relationships between them, and how the project is built using these components.
3. Distribution – Enables the cross-site synchronization of programs, and is responsible for software distribution, packing and installation.

The data dictionary defined within the CC is the source for the generation of data model upgrade scripts that are applied to the deployed databases in order to upgrade the schemas with no loss of data.

In version 6.0, the CC is composed of two main components:

- The source repository and lifecycle is Harvest
- The structure management is "Xtra-C for ccmngr".

In the CDE environment, the CC will include a product structure with components, modules and projects as it was in the original CC of ABP, except for the following changes:

- Since the ATL and INF modules are not supposed to be customized, there is no need for the CC layer to include these modules
- Each module will include 3 projects (in the original CC of ABP each module has 4 projects). The difference is the generic project that is not relevant for the customization CC.
- Each project will not include generic BBs. The CC person should create customized BB for each project the developers want to customize.

The following figure presents the CC structure of v600.

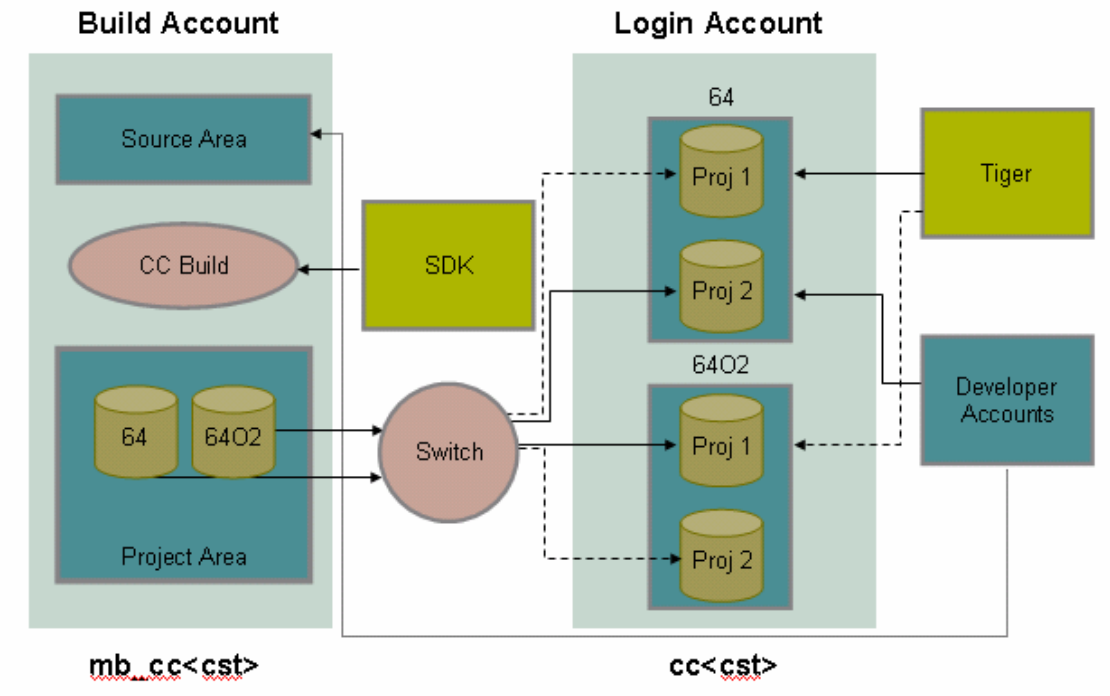


Figure 2-4: SDK Package Version 6.0 Structure

Client

Version 6.0 of the client software should be installed by the client kit in the delivery content. For installation/upgrade instruction, please refer to the ABP Client Installation Guide (CDE).

Tiger

The CDE environment includes the Tiger. The idea is that if an account has SDK and CC, it also customizes the data model objects. The SDK includes the data model definitions (sources) from the core CC and the CC layer includes the customized objects generated by the Fox. After the LEL is ready and includes customization (generated together into the proj area of the CC by the build process), the ADBA can load the objects from the CC layer into the Tiger repository.

It is highly important that both the CDE and the runtime (RT) environments will include only one Tiger repository per version. It is also recommended that the Tiger will be local on the CDE machine and by that avoid unnecessary system mounts.

Detailed Workflow and Related Documentation

The following figure presents the main stages of the upgrade workflow and the related documents or related chapters of this document where these stages are described.

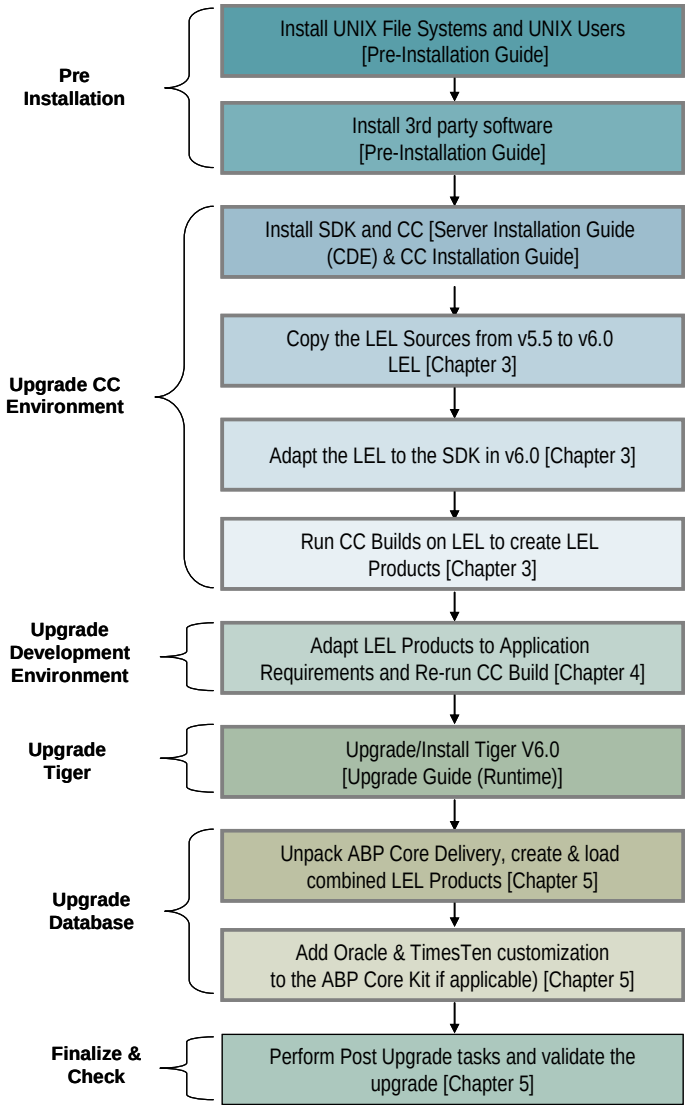


Figure 2-5: Infrastructure Upgrade Workflow

Installing Operating System

Install the required file system, UNIX and database accounts for the CDE of version 6.0. Those requirements are for installing the SDK, the CC layer and the Tiger software and repository.

The installation procedure is described in the vendor's documentation. Setting the file system, creating accounts and Configuring UNIX parameters are described in the ABP Server Pre-Installation Guide.

Installing/Upgrading Third Party Software

ABP version 5.5 requires that most third party software be upgraded to, or installed in, 64-bit mode. Most of the third-party software can be installed during the upgrade preparation phase.

The installation procedures are described in the vendor's documentation. A list of the required software and their vendors and versions appears in the ABP Server Pre-Installation Guide.

Installing SDK and CC Layer

Install the SDK and the CC Layer of version 6.0. The CC layer structure is imported from the ABP core CC and should be completed to the LEL structure the account had in version 5.5. The customized projects and building blocks the account has used in version 5.5 should be created in version 6.0. The account should create two variants in his CC layer, the debug and the optimize variant, where each one represent different build flag.

The SDK installation procedure is described in the Server Installation Guide (CDE). The CC installation procedure is described in its vendor documentation.

Copying the Sources from Version 5.5 LEL to Version 6.0 LEL

All the sources that exist in the version 5.5 LEL in the RCS should be copied into version 6.0 LEL. Make sure that all the structure in version 6.0 LEL is completed as described in the previous topic. The sources are copied from the file system into the harvest repository.

This procedure is described in Chapter 3 of this document.

Adapting the LEL Layer to the SDK

At this stage, the account should adapt the make file it had in version 5.5 to work with the SDK in version 6.0, changing paths according to the SDK structure, use ANT instead of make for the Java projects and building blocks. Additionally, application changes like interfaces and functionality are changed from version 5.5 to version 6.0 and the LEL taken from version 5.5 should be adapted to the new content.

This procedure is described in Chapter 3 of this document.

Running CC Build on the LEL

After the LEL is prepared, run CC build and create the products. The build should run twice, once in debug mode and again in optimize mode. Each mode is created on a different set of projects. The products include also the merged database SQLs (from the core and from the LEL).

This procedure is described in Chapter 3 of this document.

Adapting LEL Products to Application Requirements and Re-running CC Build

The following sections describe the changes that need to be made by the customization developers in different development areas, listed by application. Development teams only need to refer to the sections that apply to their application.

The output of this step is an upgrade kit that is ready for testing by the system test team. This kit includes all the infrastructure upgrade programs, the development customization areas and valid storage for running the tests.

This procedure is described in Chapter 4 of this document.

Upgrading the Tiger Software and Loading the Objects from the LEL

After a successful build of the GDD projects, upgrade the Tiger software, unpack the core delivery that contains default values and scripts (both steps can be done before) and load the GDD objects from the LEL.

This procedure is described in Chapter 5 of this document.

Preparing Database Upgrade

Once the GDD object of the core and the customization layer (both are merged into the LEL) are in the Tiger repository, create a database upgrade kit from version 5.5 to version 6.0. Depending on your site's architecture, the kit should be created for the Oracle and the optional TimesTen databases.

This procedure is described in Chapter 5 of this document.

Performing Post Upgrade Tasks and Validation

Perform post upgrade tasks and validate the sanity of the upgraded system.

This procedure is described in Chapter 5 of this document.

3. UPGRADING THE CC ENVIRONMENT

This Infrastructure and CC upgrade procedure provides the account Infrastructure team with step-by-step instructions on how to upgrade the Amdocs Billing Platform Customization Development Environment from version 5.5 to version 6.0.

A complete list of prerequisite software and modified hardware requirements can be found in the ABP 6.0 Server Pre-Installation Guide.

The recommended settings for important system parameters (depending on the type of hardware platform) can be found in the ABP 6.0 Server Pre-Installation Guide.

This chapter provides a comprehensive list of activities that the Infrastructure team need to perform for the upgrade. Since many activities are identical to the ones explained in the ABP 6.0 Server Installation Guide (CDE), in many cases you are referred to the relevant sections of this document. In case that the activity is not identical, this document supplies the required step-by-step procedure.

It is important to read all detailed explanations before actually performing the activity.

Main CC Differences between Version 5.5 and Version 6.0

The main differences between version 5.5 and version 6.0 regarding the development environment are:

- In version 5.5 the CC includes one Master UNIX account (cc<cst>) while in version 6.0 it includes two UNIX accounts: the login cc<cst> and the master mb_cc<cst>. The master build is running on the Master account. The Switch process copies the products from the master account to the login account. Developers account looks at the login account.
- The CC in version 5.5 includes sources of the core and of the LEL. In version 6.0, the core is SDK (products only) and the LEL remains sources.
- The repository in version 5.5 is either RCS or Xtra-C. In version 6.0 it is only Xtra-C.
- The Java build in version 6.0 is done using ANT where in version 5.5 it uses make.
- Version 6.0 supports incremental build.
- Version 6.0 is fully shared libraries.
- Version 6.0 supports both 64bit and Optimized build.

Installing UNIX File System and Users

This section lists the UNIX accounts, file systems, database instances and database users that need to be prepared by the System UNIX team and Oracle DBAs before starting the upgrade process. You can use this information as a template for your specific upgrade.

Verifying UNIX Initialization Files

1. Verify that the following lines exist in the `/etc/csh.cshrc` file on HP/AIX servers or the `/usr/local/site-init-files/site.cshrc` file on a Sun operating system:

The `/sdkhome/sdk/SDKRoot/.ccmngr_cshrc` file is delivered within the SDK installation package

```
set homedir =  `echo $HOME | cut -f3 -d/`

if ($homedir == "cde") then
   if (-f /usr/local/ccmngr/config/.ccmngr_cshrc) then
      source /usr/local/ccmngr/config/.ccmngr_cshrc
   else if (! $?CCHOME) then
      setenv CCHOME /usr/local/ccmngr/ccgnrl/v09
   endif
endif
```

2. Add the following lines to the beginning of the /usr/local/ccmngr/config/.ccmngr_cshrc file:

```
if ( ! -d /sdkhome/sdk/SDKRoot ) goto CCMNGRUSERSET

if ( -r $HOME/proj/.project.csh ) then
   source $HOME/proj/.project.csh
else
   if ( -r $CCPROJECTHOME/proj/.project.csh ) then
      source $CCPROJECTHOME/proj/.project.csh
   else
      /bin/printf "\n\t **** Error:  $HOME/proj/.project.csh NOT FOUND ...\n"
      /bin/printf "\n\tPlease run as following:\n"
      /bin/printf "\t1. source $CCPROJECTHOME/setEnv\n"
      /bin/printf "\t2. set_prod -p <proj-name> -b <bb-name> -v <variant>\n\n\n"
      exit
   endif
endif

if ( ! $?CCPRODUCTVER ) goto CCMNGRUSERSET

@ verDigits=`echo $CCPRODUCTVER | /bin/sed 's/[A-z_]//g'`

if ( $verDigits < 600 ) goto CCMNGRUSERSET

source /sdkhome/sdk/SDKRoot/.ccmngr_cshrc
```

```
exit
CCMNGRUSERSET:
```

3. Verify that the following lines exist in the `/etc/csh.login` file on HP/AIX servers or the `/usr/local/site-init-files/site.login` file on a Sun operating system:

The `/sdkhome/sdk/SDKRoot/.ccmngr_login` file is delivered within the SDK installation package

```
set homedir =  `echo $HOME | cut -f3 -d/`

if ($homedir == "cde") then
   if (-f /usr/local/ccmngr/config/.ccmngr_login) then
      source /usr/local/ccmngr/config/.ccmngr_login
   else if (! $?CCHOME) then
      setenv CCHOME /usr/local/ccmngr/ccgnrl/v09
   endif
endif
```

4. Add the following lines to the beginning of the /usr/local/ccmngr/config/.ccmngr_login file:

```
if ( ! $?verDigits ) goto CCMNGRUSERSET
if ( $verDigits < 600 ) goto CCMNGRUSERSET
source /sdkhome/sdk/SDKRoot/.ccmngr_login
exit
CCMNGRUSERSET:
```

5. Change the set_proj alias in the /usr/local/ccmngr/config/.ccmngr_aliases file.

```
alias set_proj    'setenv CCMNGRHOME "/usr/local/ccmngr";setenv CCHOMEVERSION "v09";setenv CCHOME
"$CCMNGRHOME/ccgnrl/$CCHOMEVERSION";source ~ccmngr/bin/set_proj_acm \!*'
```

6. Add the following set_prod alias in the /usr/local/ccmngr/config/.ccmngr_aliases file

```
alias set_prod   'source  $CCPROJECTHOME/setEnv;$HARCCHOME/bin/CCLockSetProj; harset_product \!*
&& source CCPROJECTHOME/data/profile.ini;$HARCCHOME/bin/CCUnLockSetProj'
```

7. ONLY in case of a Sun operating system verify that the following line exists in the ~mb_cc<cst>/.cshrc file**:**
   `source /usr/local/site-init-files/site.cshrc`
8. ONLY in case of a Sun operating system add the following line to the ~mb_cc<cst>/.login file:
   `source /usr/local/site-init-files/site.login`
9. Repeat the last two steps for the cc<cst> account.

Disk Configuration

The following file systems should exist on the server where the CDE will be upgraded.

For Content Delivery

File System	Additional Size [GB]	Description
/devdelivery600	5	

For CC and SDK

File System	Size [GB]	Description	Action	Example
/<CC_name>home	Another 5G	Disk space for CC sources	The same FS as in v5.5.	
/sdkhome	5	File system for ABP Core 6.0 SDK installation	1. Create path /sdkhome/sdk 2. Validate the size of the file system	1. mkdir –p /sdkhome/sdk 2. df –k sdkhome
/mb<Cst>proj Example: /mbcdeproj	10	File system for CC Master build account's products	1. Create path: /<MB_Cst>proj/<cst>/proj 2. Validate the size of the file system	1. mkdir –p /mbcdeproj/cde/proj 2. df –k /mbcdeproj/cde
/<Cst>proj Example: /cdeproj	20	File system for CC Login account's products	1. Create path: /<Cst>proj/<cst>/proj 2. Validate the size of the file system for the debug and optimizer products	1. mkdir –p /cdeproj/cde/proj 2. df –k /cdeproj/cde
/<Cst>user[1-N] where N – number of file systems Example: /cdeuser10	Number of users * (1 to 1.5GB)	File systems for CC developer accounts. Create new ones only if needed.	1. Create path: /<Cst>user/<cst>/<appl> 2. Create UNIX users having the same name as the NT account of the developer 3. Validate the size of the file system for each user	mkdir –p /cdeuser10/cde/bl/<NT user>

Disk Configuration for a New Tiger Installation

File System	Size [GB]	Description
/tgruser. <host suffix>	10	For new Tiger account v6.0

Disk Configuration for Upgrading Tiger Installation

File System	Size [GB]	Description
Existing Tiger FS	Enlarge by 10G	For upgrading existing Tiger installation

Oracle File Systems

File System	Size (GB)	Description
/oravl##	2	System, Temp, Undo
/oravl97	2	Oracle export files
/oravl98	2	Cold backup
/oravl99	2	Oracle archive files

TimesTen File Systems

File System	Size (GB)	Description
/ttvl02	2	

New UNIX Accounts

The following UNIX accounts should exist on the server where the CDE will be upgraded.

UNIX Accounts for Delivery Content

User Name	Used By	Home Dir	Shell	Description	Example
devupg600	CC	/devdelivery600	tcsh		

UNIX Accounts for CC and SDK

User Name	Used By	Home Dir	Shell	Description	Example
cc<cst> Example: cccde	CC	/<CC-cst>home/<cst>/cc<cst>	tcsh	Login CC Manager Account. Make sure that CRONTAB is allowed on this user – **Should be exist from v5.5**	/cdehome/cde/cccde/cccde
mb_cc<cst> Example: mb_cccde	CC	/<CC-cst>home/<cst>/cc<cst>	tcsh	Master build CC Account. Make sure that CRONTAB is allowed on this user	/cdehome/cde/cccde/mb_cccde
<NT developer-login-name>	Developer	/<CC-cst>users/<cst>/<appl>	tcsh	UNIX accounts for developers	/cdeusers10/cde/bl/yaronc

UNIX Accounts for a New Tiger Installation

User Name	Used By	Group	Home Directory	Shell	Description
dbtgr600	Tiger	adba	/tgruser.<host suffix>	ksh	

UNIX Accounts for Upgrading Tiger Installation

User Name	Used By	Group	Home Directory	Shell	Description
dbtgr	Tiger	adba	/tgruser. <host suffix>	ksh	

UNIX Accounts for TimesTen (only if needed)

User Name	Used By	Group	Home Directory	Shell	Description
Timesten	Tiger		/ttvl02/timesten	ksh	

Oracle Instances

The following Oracle accounts should exist on the server where the CDE will be upgraded.

Oracle Accounts

User Name	Used By	Description
AIM_DBA	ADBA	Tiger special account. Special privileges are required (see below)
Dbtgr** - Oracle account for Tiger installation	ADBA	Tiger Repository account. In version 600 the oracle instance should be version 9.2.0.5. This account can be used for the AMC installation.
CCAMC	CC	Username for the CC AMC monitor - Optional

Validating Oracle Accounts Privileges

10. Check that the Oracle account AIM_DBA have the correct privileges:

```
Sqlplus aim_dba/****@<instance_name>
Select * from user_role_privs
```

The expected output should be:

USERNAME	GRANTED_ROLE	ADMIN_OPTION	DEFAULT_ROLE	OS_GRANTED
AIM_DBA	CONNECT	YES	YES	NO
AIM_DBA	DEVELOPER	NO	YES	NO
AIM_DBA	RESOURCE	YES	YES	NO
AIM_DBA	SELECT_CATALOG_ROLE	NO	YES	NO

11. Select * from user_sys_privs.
    The expected output should be:

USERNAME	PRIVILEGE	ADMIN_OPTION
AIM_DBA	ALTER SESSION	NO
AIM_DBA	CREATE ANY DIRECTORY	YES
AIM_DBA	CREATE CLUSTER	NO
AIM_DBA	CREATE DATABASE LINK	NO
AIM_DBA	CREATE PROCEDURE	NO
AIM_DBA	CREATE ROLE	YES
AIM_DBA	CREATE SEQUENCE	NO
AIM_DBA	CREATE SESSION	NO
AIM_DBA	CREATE SNAPSHOT	YES
AIM_DBA	CREATE SYNONYM	NO
AIM_DBA	CREATE TABLE	YES
AIM_DBA	CREATE TRIGGER	NO
AIM_DBA	CREATE TYPE	YES
AIM_DBA	CREATE USER	NO
AIM_DBA	CREATE VIEW	YES
AIM_DBA	DROP ANY DIRECTORY	YES
AIM_DBA	SELECT ANY DICTIONARY	NO
AIM_DBA	SELECT ANY TABLE	YES
AIM_DBA	UNLIMITED TABLESPACE	YES

Installing Third Party Tools

Install the third party and Gnu software using the configurations described in the *ABP Server Pre-Installation Guide*.

Installing Oracle Instance

Install a new Oracle database instance for version 9.2.0.5 alongside the existing Oracle 9.2.0.3 instance.

If the CDE and the Runtime environments reside on different machines than you have two options:

- Create 9.2.0.5 instance on each machine. The instance on the CDE can be use for the Tiger repository and developers accounts. The instance on the RT machine will be used for RT environments.
- Create one 9.2.0.5 instance on the RT environment. This instance will serve the RT environments, the Tiger repository and private developers' accounts.

This Upgrade Guide describes how to perform a new Oracle installation. However, it should be noted that an existing 9.2.0.3 instance can be upgraded to 9.2.0.5, and the activities described for the new instance should be performed on this upgraded instance.

Upgrading the CC Environment

This section describes how to upgrade a CC development environment from version 5.5 to version 6.0.

The information in this section is only relevant if the ABP core product has been customized by an account. If no customizations have been performed, then this section can be skipped. Once the development tools have been upgraded with a new CC and Tiger, a development environment can be created.

Copying the Delivery Content

Copy the core delivery content to the devupg600 account.

File Name	Description
SDK_Server_Installation_v600_<platform>.sh	Server SDK file – Choose the one relevant for your platform
Export_<sequence>_<timestamp>.xml	CC Structure export for XtraC
Tiger_Installation_v600.tar.gz	Tiger delivery

Installing the SDK Package

Install the SDK package according to the Server Installation Guide (CDE), Chapter 4, according to the following subsections:

Section Name	Follow CDE Installation Document?	Description
Prerequisites	Yes	
Copying the SDK content	No	Already done in above subsection
Installing the SDK Package Using InstallShield	Yes	
Checking the Installation	Yes	
Post Installation	Yes	

Setting Up Xtra-C

In version 5.5, the account can have one of the three options of CC:

1. RCS without Xtra-C for ccmngr
2. RCS with Xtra-C for ccmngr
3. Xtra-C for ccmngr with Harvest.

The following subsections describe the upgrade path according to the above three options:

RCS without Xtra-C Administrator

In this case the account will install Xtra-C Administrator with Harvest version 3.0 and no v5.5 structure migration is necessary. The repository will not include version 5.5 structure.

RCS with Xtra-C Administrator

In this case the account will upgrade the Xtra-C Administrator to v3.0 with Harvest version 3.0 and no v5.5 structure migration is necessary. The repository will include version 5.5 structure.

Xtra-C Administrator with Harvest

In this case the account will upgrade the Xtra-C Administrator to Xtra-C with Harvest version 3.0 and no structure migration is necessary. The repository will include version 5.5 structure.

The link directions of data and proj directories are difernt between CDE installation and CDE upgrade. In CDE upgrade, the links are from the master to the login (mb_cc<cst> to mb<cst>) where in CDE installation it's the opposite.

After you have a ready installation of Xtra-C version 3.0 with the correct implementation, follow the steps according to Server Installation Guide (CDE), Chapter 5:

Section Name	Follow Server Installation Guide (CDE)	Description
Preparing Parameters Required for Xtra-C Installation	Yes	
Validate System Properties in Xtra-C	Yes	
Validating the set_env.setup File	Yes	
Edit the setEnv file	No	* If you did not have Xtra-C in v5.5 cc, use the instruction of the Installation Guide (CDE) for this section. * If you had Xtra-C in v5.5, merge the setEnv file definition as describe in the Installation Guide (CDE) with what you had previously.
Updating the .rhosts File	No	The .rhost file already exists from v5.5 CC. Add new entries for the new Xtra-C installation as describe in the Installation Guide (CDE) for this section.
Setting Data Directories	No	1. Enter into the mb_cc<cst> user account. 2. Create a symbolic link to the data directory on the cc<cst account: `ln -s ~cc<cst>/data ~/data`
Setting proj directories	No	See description following this section.

Setting Proj Directories

1. Login into the mb_cc<cst> master account.
2. Create links from the cc<cst> account to the mb_cc<cst> account and set it to 775 permissions for the following levels:
   - ~/product
   - ~/comp
   - ~/module
   - ~/bb
   - ~/data

   Example:

```
hpp711!mb_cccde:~ [110]> ln -s ~cccde/product ~mb_cccde/product
hpp711!mb_cccde:~ [111]> ln -s ~cccde/comp    ~mb_cccde/comp
hpp711!mb_cccde:~ [112]> ln -s ~cccde/module  ~mb_cccde/module
hpp711!mb_cccde:~ [113]> ln -s ~cccde/bb      ~mb_cccde/bb
hpp711!mb_cccde:~ [114]> ln -s ~cccde/data    ~mb_cccde/data
```

3. Create the xtrac_logs directory for logging the activities of the CC Manager GUI.

   Example:

```
hpp705!mb_cctst:~ [74]> mkdir -p xtrac_logs/module
hpp705!mb_cctst:~ [75]> mkdir -p xtrac_logs/comp
hpp705!mb_cctst:~ [76]> mkdir -p xtrac_logs/bb
hpp705!mb_cctst:~ [77]> mkdir -p xtrac_logs/product
hpp705!mb_cctst:~ [78]> mkdir -p xtrac_logs/proj
```

4. Create the configuration file that contains a list of log file masks for each build and its location `~mb_cc<cst>/data/name_mask.set_up` as follows:
   a. Login to the CC Master Build user: `mb_cc<cst>`
   b. Change directory: `cd ~/data`
   c. Copy the /sdkhome/sdk/SDKRoot/ABP600/tools/build/samples/DATA_name_mask.set_up file as follows:
      `\> cp /sdkhome/sdk/SDKRoot/ABP600/tools/build/samples/DATA_name_mask.set_up ~/ data/name_mask.set_up`
   d. Type: `\> vi name_mask.set_up`
   e. Change every `<HOME>` template string to the /<cst>home/<cst>/cc<cst>/mb_cc<cst> home directory
   f. Save and quit.

5. Create the ~mb_cc<cst>/xml/alarms.xml file. This file contains the errors to be ignored during the Harvest refresh process.
   a. Login to the CC Master Build user: `~mb_cc<cst>`
   b. Create the xml directory: `mkdir xml`
   c. Change directory: `cd ~/xml`
   d. copy the /sdkhome/sdk/SDKRoot/ABP600/tools/build/samples/XML_alarms.xml file as follows:
      ```
      \> cp
      /sdkhome/sdk/SDKRoot/ABP600/tools/build/samples/XML_alarms.xml ~/xml/alarms.xml
      ```
   e. Type: `\> vi alarms.xml`
   f. Edit the file by changing every <EMAIL ADDRESS> template string to your local email address for error notifications.
   g. Save and quit.

Section Name	Follow Server Installation Guide (CDE)	Description
Create Link to SDK	Yes	
Invoking Xtra-C Administrator's Tool	Yes	
Creating Machines	Yes	
Creating Login Profiles	Yes	
Connecting the Login and Shared Login Profiles	Yes	
Importing the LEL Product to the Xtra-C Server	Yes	
Matching the Login and Build Login Profiles to the LEL Product	Yes	
Defining the Product Variants	Yes	
Creating a Lifecycle	Yes	
Creating the Upload Package	Yes	
Running Create Repository on the LEL v600	Yes	
Running UFS on the LEL v600	Yes	
Promoting the VERSION_600 Package in Harvest	Yes	

Preparing the CC for Build

After you created the structure of the LEL product of version 600 both in the Xtra-C and on the file system, you can proceed to the next steps as describe in the Server Installation Guide (CDE), chapter 6.

Section Name	Follow CDE Installation Document	Description

Editing the .cshrc File	No	Edit the .cshrc file in the home directory for Build CC Master accounts:
		1. Login to the Build CC Master (mb_cc<cst>) user.
		2. Edit the .cshrc file (vi $HOME/.cshrc). If the file does not exist, this command creates it.
		3. Append the following lines to $HOME/.cshrc:
		if (-r ${CCPROJECTHOME}/setEnv) then
		source ${CCPROJECTHOME}/setEnv
		else
		echo "\n\t Error: ${CCPROJECTHOME}/setEnv is Miss.\n"
		endif

Creating the profile.ini File	Yes	
Creating the .SDK.setup File	Yes	
Creating the cc_local.dat.<version> File	Yes	
Creating the cc_local.dat File	Yes	
Creating Customization Files	Yes	
Creating a Customized BB for Each GDD Project	Yes	 *If you had v5.5 with Harvest that you can not use the automatic option.*
Creating Configuration Files for Every CC Project and Customize GDD Building Block	Yes	
Creating/Updating Configuration File for each customize GDD Building Block	Yes	
Generating and Running the Module Config Files	Yes	
Running Create Repository on the LEL v600	Yes	
Running UFS on the LEL v600	Yes	
Checking in the Configuration Files	Yes	

Migrating LEL Sources

After you have completed the minimal requirements for the LEL layer, you can start migrating the **customization** sources from version 5.5.

Make sure that import BB process will be used ONLY on <c[7 or 9]*> building blocks.* ***The core building blocks have to remain empty.***

Import Source Area

1. Prepare a list of building blocks in the version 5.5 LEL that includes customization codes. This list should include GDD and application building blocks.
2. Lock CC and create export for each building block in version 5.5.
3. For each building block you have in the list, create a new building block in version 6.0 (if it is GDD than it is already done):
   a. Define the new building block with the topics through the Xtra-C. For instructions on how to create Java/C++ or GDD building block, refer to Server Installation Guide (CDE), chapter 11.
   b. Run create repository
4. Run the himportbb script for all the building blocks in the list. For details, refer to the Xtra-C Administrator Guide (supplied by its vendor).
5. For each building block you migrated in version 6.0, perform the following steps:
   a. For each customization Java building block you created, do the following:
   - Convert the make.def files to ANT build files, for details refer to the Server Installation Guide (CDE), Appendix D.
   - Remove the make.def files from the harvest repository.
   - Make sure that the bb descriptor in the Xtra-C is defined as Java type.
   - If the building block is a Front End application, you need:
     - Add specific rule for the build. The build is joining the static resources from the SDK and the customization projects. For more details, refer to the Server Installation Guide (CDE) chapter 11, subsection: Creating a New FE Building Block.
     - Notice that the new build merge the static resources from the SDK and from the customization building blocks. Hence, it is unnecessary that the customization building block will include core sources (As you have in v5.5).

   b. For each C/C++ building block, adapt the make files for the CDE (see Appendix E in the Server Installation Guide (CDE)). This should include changing references to the SDK using SDK environment parameters.
   c. For each C++ building block, if you are using either **gen_cppdt** or **gen_gdddt**, you should convert them to run by Sagit. For more details, refer to the Server Installation Guide (CDE) chapter 11, subsection: Implementing C++ Generators.

d. Run UpdateFileSystem.

e. Run refresh for each customized building block.

f. In all your customization sources change the words with prefix "TLG_*","Tlg_*" and "tlg_*" to "ABP_*". Use the script replaceTlg. See usage by running with option –h.

g. Change sonar xml files on the LEL. The naming convention for all SONAR XML files has changed. The new filename includes the layer number. For example, cmbe.xml is now called cmbe1.xml for the core and cmbe9.xml for LEL.

Import Proj Area

1. You need to merge the configuration files (.project.setup, .make.def) only under the **customization** projects in version 5.5 into the customization projects of version 6.0. Note that the merge should include parameter adaptation to the SDK.
2. Check-in the updated make files.

Post Import

Generate a module.config file for each module that has a new customized building block in version 6.0 after the migration process.

Adapting Product Configuration

If you had in version 5.5 local configurations in the product level and you decide to create them in version 6.0 too, perform the following procedures:

Customizing the .product.cust.setup File

1. For customizing the generic CC setup on the product level, create the .product.cust.setup file in the ${CCPROJECTHOME}/product/${CCPRODUCT}/${CCPRODUCTVER}/config directory.

   Create this file for changing the existing setup of the .product.setup file in the $SDKROOTHOME/$SDKRELEASE/tools/build directory.

   Example: In the generic .product.setup file, the QUANTIFYHOME env variable is defined.

```
……..
if ( "${?PERFORMANCE_TOOL}" ) then
……..
 if ( "${PERFORMANCE_TOOL}" == "quantify" ) then
   source  /opt/rational/purifyplus_setup.csh
   setenv  QUANTIFYHOME  /opt/rational/releases/quantify.hp.2003a.06.00.ER.0107
   setenv  QSTUBS        -L${QUANTIFYHOME}/quantify_stubs.a
 endif
……..
Endif
```

2. The account needs to overwrite the QUANTIFYHOME environment variable. Create the .product.cust.setup file in the ${CCPROJECTHOME}/product/${CCPRODUCT}/${CCPRODUCTVER}/config directory. Use it to set the new values:

```
if ( "${?PERFORMANCE_TOOL}" ) then
……….
 if ( "${PERFORMANCE_TOOL}" == "quantify" ) then
   source  <new value>
   setenv  QUANTIFYHOME  <new value>
 endif
……..
Endif
```

Customizing Third Party Software

In case that you would like to locally change, add or delete 3rd party software installations on the product level, perform the following:

1. Create the file ${CCPROJECTHOME}/product/${CCPRODUCT}/${CCPRODUCTVER}/config/.3rdparty.cust.setup
2. Overwrite, add or unset values in this file.

The file with core setup should not be changed $SDKHOME/$SDKREALEASE/tools/build/config/.3rdparty.setup

The product is defined to compile with TimesTen as a default. If you want to change the default, do the following:

a. Change directory
   `cd ${CCPROJECTHOME}/product/${CCPRODUCT}/${CCPRODUCTVER}/`
b. Edit the .3rdparty.cust.setup file
c. Add the following line:
   `setenv WORK_WITH_TT N`
d. Save and exit.

Setting Up Local Projects Environment Parameters

For setting up local CC projects on the product level, create the file ${CCPROJECTHOME}/product/${CCPRODUCT}/${CCPRODUCTVER}/config/.projects.cust.setup

Create this file in case that either changes or a new setup are required for CC projects on the product level.

Setting Up Local Modules Environment Parameters

For setting up local CC module on the product level, create the file ${CCPROJECTHOME}/product/${CCPRODUCT}/${CCPRODUCTVER}/config/.MD.cust.setup

Create this file in case that either changes or a new setup are required for CC modules setup on the product level.

Making Local Definitions and Rules

For making local definitions and rules on the product level, create the file ${CCPROJECTHOME}/product/${CCPRODUCT}/${CCPRODUCTVER}/config/.make.def.cust

Create this file in case that either changes or new definitions and rules are required for CC build setup on the product level.

Running CC Build

After you completed the minimal requirements for the LEL layer, you can proceed to the build steps as describe in the Server Installation Guide (CDE), chapter 7.

Section Name	Follow CDE Installation document Yes/No	Description
Pre Build Actions	Yes	
Running CC Build	Yes	
Checking the Build results.	Yes	
Install AMC Monitor	Yes	

4. Upgrading the Development Environment

The upgrade of Amdocs Billing Platform (ABP) from version 5.5 to version 6.0 involves several activities that are performed by the CDE team and infrastructure team concurrently. This chapter provides a detailed view of the involved parties and the activities they need to perform.

The output of this step is an upgrade kit that is ready for testing by the system test team. This kit includes all the infrastructure upgrade programs, the development customized areas and valid storage for running the tests.

The following sections describe the changes that need to be made by the customization developers in different development areas, listed by application. Development teams only need to refer to the sections that apply to their application.

In some cases, a table summarizes all the activities to be performed and can be used as a checklist by the upgrade developers.

CDE Upgrade Preparation

The following table lists the steps that need to be performed by the CDE teams when preparing the version 6.0 core modules:

Activity Item	Activity Description	Scheduling	Automatic Process	Prerequisite for development	Comments
Upgrade 3rd parties	Upgrade / install new 3rd parties, according to the platform used.	Platform installation	No	Yes	Developers should use the updated third-party packages.
Amdocs dStudio upgrade	Upgrade to 2.2.9	Pre development	No	Yes	The platform is used by the Fox, therefore it should be upgraded
Fox upgrade	Upgrade to 2.2.9. New features include extending domain valid values, table extensions, new DB objects, and selection of specific core indexes.	Pre development	No	Yes	
Use Fox domain valid values extension	Delete all old core domains in customization area, add all domains VVs via fox	Development	No	No	

Activity Item	Activity Description	Scheduling	Automatic Process	Prerequisite for development	Comments
Deprecated APIs removal	Adjust the any code that uses a deprecated API, that has been removed in the new version	Pre build	No	Yes	Running the build process is a simple way to find these removed APIs
Use updated third-party software	Use the updated third-party software that applies to this version.	Pre build	No	Yes	Required for first stable CC build
Full build	Full build of the customization BBs		Yes	Yes	
Stop using deprecated APIs	Several APIs are declared as deprecated, and may be removed in the next release. This step, is optional, involves with stop using these APIs, and start using the alternatives	Development	No	No	Performing this step will save the need to handle "deprecated APIs removal" step on the next version
Modify changed APIs					

Audit & Control (A&C)

Summary of Upgrade Activities

The following table summarizes the activities that participate in the development upgrade process. The activities are presented in the recommended order of their execution. Further details about each activity are described in subsequent sections.

#	Activity	Description	Scheduling	Automatic Process	Prerequisite for Development	Comments
1	Adjust Application Make Files Headers	Adjust the API's headers used by customization.	Build	No	No	
2	Regenerate Domains	Regenerate all customized domains	Build	Yes	Yes	
3	Modify Changed Database Structures					
4	Stop Using Deprecated APIs	Several APIs are declared as deprecated, and may be removed in the next release. This step is optional, if it is required to stop using these APIs, and start using the alternatives	Development	No	No	Performing this step will save the need to handle the "deprecated APIs removal" step in the next version
5	Adjust Batch Processes	Integration with SONAR. Adjust the modified core's scripts Operational setup	Development	No	Yes	
6	Update Configuration Changes	In case that the PC CM Load configuration files were customized in the previous version, merge the changes with the new files, delivered with this version	Development	No	Yes, if the files were previously customized	
7	Update Customization Interface	Describes enhancements that were done in customization interfaces in the scope of version 6.0.	Development	No	Yes	

Adjust Application Headers

Adjust the A&C C APIs headers that were changed due to the sourceless distribution concept. The deleted headers are described together with their substitution of new headers.

- **Ac1Interface_main.h** – for main A&C flow – will include headers of the main APIs that are used by all the applications:
  - ac1creout_io.h
  - ac1endinp_io.h
  - ac1startpgm_io.h
  - ac1endpgm_io.h
  - ac1endinpbal_io.h
  - ac1getinp_io.h
  - Relevant API's prototypes from GenAC1NRT.h.
- **Ac1Interface_cust.h** – for customized layer
- **Ac1Interface_misc.h –** miscellaneous flow (not main and not physical) – will include headers of the miscellaneous APIs that are used by all the applications:
  - ac1chkin_io.h
  - ac1getgroup_io.h
  - ac1getinfo_io.h
  - ac1newfil_io.h
  - ac1updfilests_io.h
  - ac1updrecs_io.h
  - ac1updstscn_io.h
  - ac1updwr_io.h
  - ac1updrecovery_io.h
  - Relevant API's prototypes from GenAC1NRT.h
- **Ac1Interface_physical.h** – for physical (A&F) A&C flow – will include headers for APIs that are used mainly by A&F application:
  - ac1addmisseq_io.h
  - ac1chkmisseq_io.h
  - ac1defphy_io.h
  - ac1dirlist_out.h
  - ac1endlogic_io.h
  - ac1endphy_io.h
  - ac1getoutinfo_io.h
  - ac1getouttp_io.h
  - ac1getseqno_io.h
  - ac1newphy_io.h

- ac1updouttbl_io.h
- ac1updsrc_io.h
- Relevant API's prototypes from GenAC1NRT.h .

Stop Using Application Internal Domains

Due to the sourceless concept, each application publishes the relevant headers for using its APIs. All uses of the internal A&C domains and structures should be removed.

Modify Changed Database Structures

Database changes are described in the A&C Release Notes. The automatic database upgrade process is responsible for the structure and data upgrade. The data upgrade is required when there is a need to upgrade a live environment containing data from 5.5 to 6.0.

The Tiger tool supports running the upgrade.

Stop Using Deprecated APIs

The following C APIs were declared as deprecated in this version and should not be used in the long term. As deprecated APIs can be removed from future versions, it is highly recommended to stop using these APIs.

The alternative is to move to C++ or Java new APIs..

API Name
ac1_start_program
ac1_end_program
ac1_get_next_file
ac1_create_output
ac1_end_input_balance
ac1_end_input_update
ac1_new_file
ac1_upd_recs
ac1_upd_wr
ac1_upd_sts_cn
ac1_chk_in
ac1_upd_file_sts
ac1_get_info
ac1_upd_recovery
ac1_get_next_group
ac1_new_physical
ac1_get_output_info
ac1_update_output_seq
ac1_update_output_table
ac1_end_phy
ac1_get_seq_no

API Name
ac1_add_mis_seq
ac1_chk_mis_seq
ac1_upd_src_seq
ac1_get_out_type
ac1_dir_list
ac1_end_logical

Adjust Batch Processes

The following batch A&C processes are not in use anymore:

- AC1AuditMonitorDaemon
- AC1MoveToCleanup
- AC1CleanupMain

All the customization related to these processes should be upgrade. The extension of the A&C tables should be improved by the A&C customization guide document. The cleanup should be configured in the generic cleanup process.

Update Configuration Changes

This section details all configuration changes that were made in the core layer that impact the customization layer. Configuration properties are maintained in properties files and other various configuration files.

Properties

The AC1_ACTIVITIES_CUST table that includes all the C APIs customization methods should be updated with the relevant activities from AC1_ACTIVITIES.

Update Customization Interface

The customization concept relating to the extension of A&C tables was changed. The old concept says to extend the original table. The new concept is to define a customized table with the same primary key as the A&C original table. The customization gets the key as parameter and uses it for different activities on customized table. More details will be found in the customization guide.

Customization Public Objects

The header for customization that includes all the definitions for exit points argument structures and statuses is **Ac1Interface_cust.h.**

Exit Points Changes

List of removed deprecates APIs exit points:

Exit Point Activity Name
A_GNF_HNDL_CUST_DATA
A_EIB_MORE_BALANCE

Exit Point Activity Name
A_EIU_CALL_STAT
A_NPY_EXTRA_VALID_CHK
A_DPY_EXTRA_VALID_CHK
A_EPY_EXTRA_ACTION
A_CMS_EXTRA_VALID_CHK
A_CMS_HNDL_CUST_DATA
A_AMS_EXTRA_VALID_CHK
A_AMS_HNDL_CUST_DATA
A_GSN_HNDL_EXTRA_DATA
A_GOI_HNDL_EXTRA_DATA
A_UOS_EXTRA_VALID_CHK
A_UOS_HNDL_CUST_DATA
A_UPS_HNDL_CUST_DATA
A_UOT_EXTRA_VALID_CHK
A_UPR_EXTRA_VALID_CHK
A_UWRQ_EXTRA_VALID_CHK
A_GNG_HNDL_CUST_DATA
A_UST_EXTRA_VALID_CHK
A_UST_HNDL_CUST_DATA
A_CHI_EXTRA_VALID_CHK

List of changed deprecates APIs exit points (arguments have been changed):

Exit Point Activity Name
A_CRO_HNDL_CUST_DATA
A_DPY_HNDL_CUST_DATA
A_DPY_MORE_TABLE_CHK
A_EIU_HND_CUST_DATA
A_GNF_GET_CUST_DATA
A_GNG_GET_CUST_DATA
A_NPY_HND_CUST_DATA
A_UPR_HNDL_CUST_DATA
R_CRO_RCP_DECISION
R_GNF_CHK_IF_RCP
R_GNG_CHK_IF_RCP

List of new deprecates APIs exit points:

Exit Point Activity Name
A_EPY_HNDL_CUST_DATA
A_NFL_HNDL_CUST_DATA

Acquisition & Formatting

Summary of Upgrade Activities

The following table summarizes the activities that participate in the development upgrade process. The activities are presented in the recommended order of their execution. Further details about each activity are described in subsequent sections.

#	Activity	Description	Scheduling	Automatic Process	Prerequisite for Development	Comments
1	Modify Changed Database Structures	Database changes are described in the A&F Release Notes. The automatic database upgrade process is responsible for the structure and data upgrade.	Run Time	Yes	Yes	
2	Stop Using Obsolete Tables	Database changes may include tables that become out of use in the scope of version 6.0. This activity may have an impact on the customization layer.	Development	No	Yes	
3	Stop Using Deprecated APIs	Several APIs are declared as deprecated, and may be removed in the next release. This step is optional, if it is required to stop using these APIs, and start using the alternatives	Development	No	No	Performing this step will save the need to handle the “deprecated APIs removal” step in the next version
4	Modify Changed APIs	Several APIs were modified in the scope of version 6.0 and as a result few enhancements should be done in the customization layer.	Development	No	Yes	
5	Adjust Batch Processes	Describes enhancements that should be done in the customization layer for specific implementations.	Development	No	Yes	
6	Update Configuration Changes	This section details all configuration changes that were made in the core layer that impacts the customization layer.	Development	No	Yes	
7	Update Customization Interface	Describes enhancements that were done in customization interfaces in the scope of version 6.0.	Development	No	Yes	

Modify Changed Database Structures

Database changes are described in the A&F Release Notes. The automatic database upgrade process is responsible for the structure and data upgrade. The data upgrade is required when there is a need to upgrade a live environment containing data from 5.5 to 6.0.

The Tiger tool supports running the upgrade.

Stop Using Obsolete Tables

In this version the following tables were replaced:

Table Name	New Table to be Used
MI1_INCOL_SID	MI1_INCOL_PROVIDER
MI1_SID	MF1_SID
MI1_SPECIAL_NUMBERS	MI1_SPECIAL_RESOUCRE

The MI1_SPECIAL_NUMBERS table is declared as deprecated in this version and should not be used in the long term. As deprecated table can be removed from future versions, Amdocs highly recommends to stop using this table.

Stop Using Deprecated APIs

The following APIs were declared as deprecated in this version and should not be used in the long term. As deprecated APIs can be removed from future versions, it is highly recommended to stop using these APIs.

API Name	Alternative
GGD_gd_GUIDE_api_event_rule_approach	GGD_gd_GUIDE_api_init
GGD_gd_GUIDE_api_init_cycle_state_buff	GGD_gd_GUIDE_api_init

Modify Changed APIs: Guiding

The Guiding APIs were changed in the scope of version 6.0 due to the following requirements:

- Sourceless – handling the extension mechanism of the Guiding Data Model
- Flexible Bill Cycle – additional input/output attributes
- Resource Ranges – additional core table in the Guiding area

Initialization API

All initialization procedures were conducted to a single API (GGD_gd_GUIDE_api_init), which is responsible for the following activities:

- Get Event Rule information
- Initialize the Cycle State Buffer
- Load TimesTen library if necessary
- Open OCCI connection
- Load and parse the XML according to DB Type
- Initialize and register the activities

The previous APIs that dealt with initialization as separate procedures are still functional, but it is strongly recommended to use this API.

Closing OCCI Connection API

A new API has been exposed (GGD_gd_GUIDE_api_close_oraConnection) for closing the OCCI connection in the execution termination stage.

Main Guiding API

The input structure of the main guiding API (GGD_gd_GUIDE_api_guiding_to_subscriber) was changed to include:

- Both regular and resource ranges queries strings
- An indication to determine which query to use
- A pre-fetch row count, which determines the number of rows OCCI fetches from Oracle for each DB access.

The output structure of the same API was changed to include:

- The scope ID retrieved from the GD1 RESOURCE_RANGES table
- The source/target customers partition ID
- Effective/expiration dates from the GD1 SUBSCR KEY and GD1 SUBSCR DEST tables.

When the Partition ID is not found, the Guiding API fails with special return code (“109”).

Modify Changed APIs: Distribution

- All RBMS rules that call a "distribute" user function in order to distribute <USAGE> events to the Rater must use the "distribute600" user function and must send to the user function the Partition ID that was received from Guiding when the event was guided, as the parameter: Data Group Body.
- All RBMS rules that call a "distribute" user function in order to distribute Guiding Error events to the Raters' Dispatcher process must use the "distribute600" user function and must send to the user function the <Guiding Errors Partition ID> as the parameter: Data Group Body.
- The parameter – Data Group Body that is sent from the RBMS rules to the "distribute600" user function must be a Working Memory field (not a constant string) with working memory length of exactly 30 bytes. This restriction exists due to technical problems related to the backward compatibility of the multiple "distribute" user functions.

Modify Changed APIs: Mapping

While mapping the cycle code of the <USAGE> event, if a guiding failure occurred, the appropriate guiding errors cycle code should be mapped using the RBMS rules. The same value needs to be populated in the corresponding configuration parameter DTF_CYCLE_CODE

Adjust Batch Processes

Since the production of High Usage Report was excluded from the main flow of the Outcollect Prepare process, the Main Driver using RBMS rules is responsible for generating a flat file for the File To Database process. This file is created from a set of mapping rules (USAGE format to INFO format). The FTD process loads the flat file to the MF1 OUTCOL RATED EVNET table for generating the new report using the Extract Tool.

The following table describes the new layout and mapping rules that should be implemented in the customization layer:

Field Name	Type	Mapping
Period	Integer 4	
Event ID	Alphanumeric 18	Usage Event ID
Cycle Month	Integer 2	Usage Cycle Year
Cycle Year	Integer 2	Usage Cycle Month
Status	Alphanumeric 1	"D"
File Name	Alphanumeric 50	
Duration	Integer 4	
Imsi	Alphanumeric 15	Usage IMSI
SDR Amount	Integer 8	Usage Charge Amount
Provider ID	Alphanumeric 6	
Start Time	Alphanumeric 14	Usage Start Time

The layout should be marked as "Integer/Float Alignment".

Update Configuration Changes

This section details all configuration changes that were made in the core layer that impact the customization layer. Configuration properties are maintained in properties files and other various configuration files.

Properties Files

- The XML configuration file (uh_FL_uh_ParamsForUh.xml) which includes all configuration parameters of the UH processes is replaced in version 6.0 by the new configuration file (UHParamsConfig.xml). The new file should be loaded to the GD1_XML_CONFIG table instead of UTL1_FILE_LOADER (SCHEMA should be populated as gd1UHParamsConfig).
- Additional XML files (UHUnloadQueries.xml, UHLoadQueries.xml, UHErrHandling.xml, GDGuidingQueries.xml) should be loaded into the GD1_XML_CONFIG table for the purposes of the Guiding API and UH processes (SCHEMA should be populated as gd1UHUnloadQueries, gd1UHLoadQueries, gd1UHErrHandling, gd1GDGuidingQueries correspondingly).
- Additional XML files for the purposes of the Multi-Threaded Incremental Unload should be loaded into the GD1_XML_CONFIG table:
  - gd1ExtractCustomerDetailsByCustomer.xml
  - gd1ExtractCustomerDetailsByOneCustomer.xml
  - gd1ExtractEdValueBySubscriber.xml
  - gd1ExtractEventRule.xml
  - gd1ExtractOnlySubscriberParameters.xml
  - gd1ExtractParamProperties.xml
  - gd1ExtractResourceRangesBySubscriber.xml
  - gd1ExtractResourceRangesNoHistory.xml
  - gd1ExtractSubscriberParametersBySubscriber.xml
  - gd1ExtractSubscriberParametersNoCustomer.xml
  - gd1ExtractSubscriberResourcesBySubscriber.xml
  - gd1ExtractSubscriberResourcesNoHistory.xml

The SCHEMA should be populated as gd1CMExtractEngineScript for all the above entries.

  - gd1CMGenericExtractConfigurationFile.xml
  - gd1CMGenericExtractTRBTransactionsFile.xml

The SCHEMA should be populated as gd1CMExtractConfig for both of the above entries.

- The XML files (uh_FL_uh_Register_UH_Full.xml and uh_FL_uh_Register_UH_Load.xml which retrieves and publishes TRB transactions) should be loaded to UTL1_FILE_LOADER for the purpose of the Incremental Unload and Load processes.

Customers who do not perform any customizations in the Guiding and Update Handler core functionalities can use the core implementation and load the core XML files using the script Gd1XmlConfigLoader located in cmf600 proj area.

Other Configuration Files Changes

The updated A&F XML configuration parameter file (mf_FL_mf_config_param.xml) should be loaded into the UTL1_FILE_LOADER for the purpose of the A&F processes.
A detailed list of all changes that were done in this configuration parameters file can be found in the A&F Release Notes document.

Update Customization Interface

The implementation method of extending the Guiding data model was changed in version 6.0. A detail description of the customization steps is included in the Guiding Customization Guide.

Exit Points Changes

All new exit points that were added in version 6.0 to the core product and all existing exit points that were changed are detailed in the A&F Release Notes document.

Application, Monitoring & Control Plug-ins

All AMC upgrade procedures refer to the installation of AMC version 6.0 in parallel to the existing one (version 5.5).

Adding New Configuration Files to the AMC

Target

The version 5.5 environment contains configuration files which are not part of the AMC version 6.0 installation.

Procedure

1. From the config directory of the AMC new installation, run AmcProfile.
2. From the command line perform **Run AmcLoadFile <File Name>** for each file. AMC will automatically convert the file to the new format.
3. The action will result in one of the following messages:
   - In case of success: “The File was Converted From Amc V5x to Amc V60”
   - In case of failure: “This file is Not according to the DTD”. Additional information regarding the failure is provided.
4. Restart AMC

Upgrading AMC Security

Target

Converting Users from the version 5.5 installation into the new security model (UAMS) provided in version 6.0.

Procedure

1. From the config directory of the AMC new installation, run AmcProfile.
2. Make sure AMC is down.
3. From the command line run: **AmcUamsDBConverter <user> / <password>**. The user (and password) must be authorized to edit UAMS security.
   During the installation, AMC is supplied with a default user named 'Admin' and password 'Admin11'.
4. The action will result in:
   - A log file, named AmcUamsDBConverter.log is created after every session. This log describes in detail the entire process flow.
   - The Converter also solves restrictions occurring in UAMS (the password must consist of upper-case letters, lower-case letters, and numerals). If a password (drawn from the database) is not valid (according to UAMS restrictions), the default value of unix11 is used instead.
5. Start AMC

Updating Existing AMC Configuration Files

Target

The version 5.5 Environment consists of modified configuration files, which exist also in the AMC new installation.

Procedure

1. Open any software editor that compares documents (XmlSpy is preferred). Update the AMC version 6.0 configuration files with the changes made in version 5.5.
2. Run AmcLoadFile to the updated file.
3. Restart AMC.

Billing

Summary of Upgrade Activities

The following table summarizes the activities that participate in the development upgrade process. The activities are presented in the recommended order of their execution. Further details about each activity are described in subsequent sections.

#	Activity	Description	Scheduling	Automatic Process	Prerequisite for Development	Comments
	Regenerate Valid Values	Delete all old core domains in the customization area, add all domains VVs via Fox	Development	No	No	The old solution is applicable, but not recommended
	Adjust Application Make Files	Adjust the customized make files to the new version	Pre Build	No	No	
	Regenerate Domains	Regenerate all customized domains	Build	Yes	Yes	
	Regenerate Datatypes	Regenerate all customized datatypes	Build	Yes	Yes	
	Modify Changed Database Structures					
	Stop Using Deprecated APIs	Several APIs are declared as deprecated, and may be removed in the next release. This step is optional, if it is required to stop using these APIs, and start using the alternatives	Development	No	No	Performing this step will save the need to handle the “deprecated APIs removal” step in the next version
	Modify Changed APIs					
	Update Configuration Changes	In case that the PC CM Load configuration files were customized in the previous version, merge the changes with the new files, delivered with this version	Development	No	Yes, if the files were previously customized	

Regenerate Valid Values

All customized FOX objects need to be regenerated prior to performing the upgrade.

Adjust Application Make Files

The following table summarizes the customized Make file changes that are relevant for version 6.0.

Modify Changed Database Structures

Database changes are described in the Billing Release Notes. The automatic database upgrade process is responsible for the structure and data upgrade. The data upgrade is required when there is a need to upgrade a live environment containing data from 5.5 to 6.0.

The Tiger tool supports running the upgrade.

As database objects may be used by customizations (e.g., map customization views to core table fields), and removed or modified objects may impact the customization layer, the customization layer should check its customized objects with the database changes listed in the release notes, and modify them according to core changes.

Adjust Daemon Processes

Load Balancer Processes

1. Use BC to redefine the processes.
2. Copy the contents of the Billing.xml.ksh file and paste it into the file in CC (bb – gamcc or AMC customization building block, under bl/src). The file can be found in BL1_XML_CONFIG with the following SCHEMA: 'AMC_BILLING'.

Daemon Processes:

The addition of daemon processes has changed for version 6.0. The configuration file has been replaced with OP scripts that insert data to the GN_TASK_CONNECT tables and create the daemon configuration from the templates of all relevant applications. The following should be done to add daemons:

1. Update the following files in CC under bb: gamcc or the AMC customization building block under /bl/src:
   a. BL[LAYER]_CoreDmnData.sql: Add insert command for
   INSERT INTO GN1_TASK_CONNECT (TASK_NAME, DB_CODE, RUN_MODE, SESSION_ARG, SYS_CREATION_DATE, SYS_UPDATE_DATE, OPERATOR_ID, APPLICATION_ID, DL_SERVICE_CODE, DL_UPDATE_STAMP, CONNECT_CODE) VALUES ('[PROCESS_NAME]', 'M3G', 'F', 'DEFAULT', SYSDATE, NULL, NULL, NULL, NULL, NULL, 'DMN0001BL1HOS');

b. BL[Layer]_Daemon_TMPL.xml.ksh: Replace the following: [PROCESS_NAME], [START_COMMAND}, [COMMAND_PARAMETERS] ,[NAME EXE], [NAME FLOW], [PROCESS_INSTANCE]

```
<Daemon UniqueResourceName="[PROCESS_NAME]">
<Description> Billing Transaction Listener for SOR</Description>
<LogFile> ${ABP_BL_ROOT}/log/*[PROCESS_NAME]*.log </LogFile>
<Operation EnableKill="No" User="#USER#">
<MroID ProcessName="[PROCESS_NAME]"
InstanceName="[PROCESS_INSTANCE]">#MRO_BL#</MroID>
<Start>
<Command>[START_COMMAND]</Command>
<Parameters>[COMMAND_PARAMETERS]</Parameters>
</Start>
<Stop>
<Command>
bl1_killproc_Ksh #USER# [STOP PARAMETERS]
</Command>
</Stop>
<SearchCriteria>
<FindExe ExeName="[NAME EXE]" MatchExact="Yes">
<Parameters>[COMMAND PARAMETERS]</Parameters>
</FindExe>
</SearchCriteria>
</Operation>
<UserDefinedColumns>
<HeaderData Value="[NAME FLOW]" id="Flow Name"/>
</UserDefinedColumns>
</Daemon>
```

2. Update Billing.xml.ksh:

a. Search for the following "<Name>Monitor & Control</Name>" and choose either the Ongoing or the Cycle Operations node. Add the unique resource name of the daemon to the filter of the command. Example:

```
<Command Type="javaClass">AmcFilterPlugIn.run(AmcDmn_${USER}_output.xml,Headers="DaemonName $
DaemonDescription $ Host $ User $ StartTime $ Status $ Flow Name $ Pid " ,RowsPrefix="DaemonName-
BLBTLSOR | BLBTLEOD | BLBTLQUOTE | ORCPCE | ORCPCA | ORCBDI | ORCBDE | CHGCYC |
COMWTDB | NEW_PROCESS_NAME ")</Command>
```

b. Update the apiInput for the daemons:
Add the following to the "processes" node under the ongoing or cycle node, depending on the process, and replace the [NAME] and [INSTANCE]. If the process is running on a different MRO, the ${MRO_ID1} must also be replaced.

Update Configuration Changes

This section details all configuration changes that were made in the core layer that impact the customization layer. Configuration properties are maintained in properties files and other various configuration files.

Properties Files

The properties file BL1amdocs_bl3g.properties was divided to three files, as follows:

- BL1JF.properties contains all the properties that BL supplies to the Java Foundation regardless of the run mode (EJB or Batch).
- BL1JFEJB.properties contains all the properties that BL supplies to the Java Foundation that is specific to the EJB run mode.
- BL1App.properties contains all the properties that BL needs regardless of the run mode.

The names of the majority of the properties were changed. For example: “amdocs.bl3g.loghandlers” is named now : “amdocs.mhs.loghandlers”. No layer should directly query a property by its name, only through a class API, so no problem should rise. If, however , the customization layer did query directly by property name, its code should be fixed.

Other Configuration Files Changes

The files jdbcProperties.xml & SONARjdbcPool.xml were deleted. There is no longer a need to use a JDBC Pool directly from the Billing APIs. If the customization layer used it directly, it should upgrade its code to use the core data sources.

Update Customization Interface - Exit Points Changes

New Events in the Documentation Flow

- **setPrintingCategory** – allows changing the printing category.
- **getBillDateForDueDate** – determines the bill date and due date for statements
- **startGroup** – gives the customization context of group plus the input files to allow low bill functionality.

New Event in the RcProration Flow

- **updateCycleType** – indicates the customization type of the cycle being processed (Frequency + Multiplier)**.**
- **getExchangeDate** – get currency exchange date. Used in case RC credit should be generated and charge currency being credited differs from discount or AR charge adjustment currencies.

CycleContextInfo – Not Backwards Compatible

All the events that received ContextInfo datatype will now be receiving CycleContextInfo datatype that will hold the same attributes except for *Cycle Month*, which was changed to *Cycle Instance*.

The affected events are:

- openCharges
- setContextInfo
- undoCustomer
- getExchangeDate
- getBillDateForDueDate (in BfInitiator flow)
- createChargeRequests
- sendSubscriberCharge
- getBillDateForDueDate (in BfInitiator flow)

Compatibility issues:

- Requires code changes – version 5.5 implementation should get the new datatype (any reference to cycle month should be changed to cycle)
- Any use of the getMonth method should be replaced with getInstace method

ProrationParamsInfo – not backwards compatible

All the events that received BasicProrationParamsInfo datatype will now be receiving ProrationParamsInfo datatype that holds more pro-ration info.

The affected event is *getProrationFactor*.

Compatibility issues:

- Requires code changes – version 5.5 implementation should get the new datatype (no other change is required)

CyclePayerInfo – Not Backwards Compatible

All the events that received CyclePayer class will now be receiving CyclePayerInfo datatype that holds more Payer Info (most of the fields in the bl1_blng_arrangement and bl1_cyc_payer_pop tables).

The affected event is *setDropBill*.

Compatibility issues:

Requires code changes – version 5.5 implementation should get the new datatype (no other change is required)

CM Online

The upgrade of Amdocs CM Online from version 5.5 to 6.0 consists of re-installing the client application. For details, see the Amdocs Billing Platform Client Installation Guide.

Customer Management

Summary of Upgrade Activities

The following table summarizes the activities that participate in the development upgrade process. The activities are presented in the recommended order of their execution. Further details about each activity are described in subsequent sections.

#	Activity	Description	Scheduling	Automatic Process	Prerequisite for Development	Comments
1	Install / upgrade all 3rd parties	Follow version 3rd party matrix, and install / upgrade all 3rd parties.	Development	No	No	
2	Adjust Application Make Files	Adjust the customized make files to the new version	Pre Build	No	No	Version 6.0 has migrated to use ANT scripts instead of make files. It's very recommended for the LEL to adopt this methodology
3	Regenerate Domains	Regenerate all customized domains	Build	Yes	Yes	
4	Regenerate Data Types	Rebuild the customization area	Build	Yes	Yes	
5	Change DVM files format	Run DVM migration script	Development	No	Yes	The DVM is now part of the studio, the format change is mandatory to comply with the studio standard. See Appendix B: DVM, RDM Persistence Tools - Migration Script
6	Regenerate Database Views	Regenerate all customized views	Build	Yes	Yes	
7	Migrate Reference datalayers to DVM	Run DVM migration script	Development	No	Yes	. See Appendix B: DVM, RDM Persistence Tools - Migration Script
8	Regenerate Reference Data Layers	Regenerate all customized data layers	Build	Yes	Yes	
9	Modify Changed Database Structures	Make DB & software adaptations according to the new structure	Development	No	No	
10	Remove APIs	Adjust any code that uses a deprecated API that has been removed in the new version	Pre build	No	Yes	Running the build process is a simple way to find these removed APIs

#	Activity	Description	Scheduling	Automatic Process	Prerequisite for Development	Comments
11	Stop Using Deprecated APIs	Several APIs are declared as deprecated, and may be removed in the next release. This step is optional, if it is required to stop using these APIs, and start using the alternatives	Development	No	No	Performing this step will save the need to handle the "deprecated APIs removal" step in the next version
12	Update Operational Changes	Adjust to new OP map	Operational setup	No	No	
13	Integrate with SONAR	Modify customization configuration files to use the SONAR standard	Development	No	No	
14	Merge Batch / Daemon execution scripts	In case core process were extended, on the execution script layer, in previous versions, merge these changes into the 6.0 execution scripts	Pre build	No	No	
15	Update CM1XMLOAD Configuration Changes	In case that the PC CM Load configuration files were customized in the previous version, merge the changes with the new files, delivered with this version	Development	No	Yes, if the files were previously customized	
16	Update XML API XSDs	Generate new APIs	Development	Yes	No	In case there are Customization APIs, and they are needed to be invoked asynchronously by the APInvoker, they should be regenerated
17	Adjust implementation core data	Merge core's new impl data, with customizations. Relevant tables: CSM_ACTIVITY CSM_ACT_RSN CSM_FLOW_CUSTOMIZE MEMO_TYPE (see note)	Implementation	Partial	Yes	This is an overall effort that should take place, in adjusting LEL customization to new 6.0 core's implementation

#	Activity	Description	Scheduling	Automatic Process	Prerequisite for Development	Comments
18	Merge published XSDs	Merge any changes that were added to publish XSDs to core's new XSDs	Development	No	No	

The Memo mechanism has been extracted from CM, to a generic component. The table MEMO_TYPE is not needed anymore. An upgrade script loads the data from the old table to the new one, and update CSM_ACT_RSN to point to the new types. However, in case of core implementation if new memos are needed, they should be manually loaded into the tables, after this upgrade

Regenerate Objects

The following **customization** objects must be upgraded and regenerated:

- Domains
- Datatypes
- DVM and RDM views and data layers

DVM and RDM source files should be upgraded. See Appendix B: DVM, RDM Persistence Tools - Migration Script

Adjust Application Make Files

Apache Ant

In version 6.0 the ANT utility is used instead of the make files in order to build the CM component. Due to this change the Sonar template builder is called as part of the Ant build.

The Build process is the first phase of creating the final product out of the application sources. The input to this process consists of the application sources and the output consists of application products with simple packaging utilities.

The Make files are no longer used by the Customer Management component to perform the build. They have been replaced by XML files used by a new tool called Apache Ant. Apache Ant is a Java-based tool used to develop software across multiple platforms and to create the final product out of the applications sources.

Modify Changed Database Structures

Database changes are described in the Customer management Release Notes. The automatic database upgrade process is responsible for the structure and data upgrade. The data upgrade is required when there is a need to upgrade a live environment containing data from 5.5 to 6.0.

The Tiger tool supports running the upgrade.

As database objects may be used by customizations (e.g., map customization views to core table fields), and removed or modified objects may impact the customization layer, **these changes are listed here in addition to the Release Notes.**

DB Object Area & GDD BB	Object Name	Object Type	Object Change Type	Column Name	Column Change Type	Domain	Change Description
Gcmgdd REF	CSM_ACTIVITY	Table	Update	ACTIVITY_GROUP_ID	New	CM1ACTGRPID	This column defines an activity group code which is common to several activity paths. It is used to map a number of activity paths to a single activity group. This column's value is used later by the Billing Component, in order to filter the relevant CM TRB transactions. This field is populated by the ABP core implementation team, for all the core implementation supported activity paths. All customized paths should be populated accordingly, and if a new group ID is added, it must be configured in the Billing component too.
Gcmgdd APP	SUBSCRIBER	Table	Update	SUB_STS_LAST_ACT	New	CM1STSACT	The last activity that caused the subscriber status modification. Values are taken from the CM Activity reference table. Was populated from the field activity_code (in the csm_activity table). Now should be populated from the corresponding field activity_group_id (in the csm_activity table)
Gcmgdd APP	SUBSCRIBER_HISTORY	Table	Update	SUB_STS_LAST_ACT	New	CM1STSACT	The last activity that caused the subscriber status modification. Values are taken from the CM Activity reference table. Was populated from the field activity_code (in the csm_activity table). Now should be populated from the corresponding field activity_group_id (in the csm_activity table)
Gcmgdd APP	SERVICE_AGREEMENT	Table	Update	INIT_ACT_CODE	New	CM1STSACT	Represents the activity code, which caused the addition of the specific offer. Should be the activity_group_id (in the csm_activity) which inserted the record. No way of knowing this information

DB Object Area & GDD BB	Object Name	Object Type	Object Change Type	Column Name	Column Change Type	Domain	Change Description
Gcmgdd APP	SERVICE_AGREEMENT	Table	Update	SOC_STS_LAST_ACT	New	CM1STSACT	The last activity that caused the offer status modification. Was populated from the field activity_code (in the csm_activity table). Now should be populated from the corresponding field activity_group_id (in the csm_activity table)
Gcmgdd APP	SRV_AGR_HISTORY	Table	Update	INIT_ACT_CODE	New	CM1STSACT	Represents the activity code, which caused the addition of the specific offer. Should be the activity_group_id (in the csm_activity) which inserted the record. No way of knowing this information
Gcmgdd APP	SRV_AGR_HISTORY	Table	Update	SOC_STS_LAST_ACT	New	CM1STSACT	The last activity that caused the offer status modification. Was populated from the field activity_code (in the csm_activity table). Now should be populated from the corresponding field activity_group_id (in the csm_activity table)
Gcmgdd REF	SUB_PROD_TYPE_REL	Table	Deleted				The table Agreement Product Type Relation replaces this table, since it should hold both subscriber types and agreement types
Gcmgdd APP	CUSTOMER	Table	Update	PARTITON_ID	New	Cm1custpart	A number representing the ID of the partition. The field have to be populated with the cycle code from the CM1_CYCLE_PARTITION_REL table
Gcmgdd APP	CUSTOMER	Table	Update	NEW_PARTITION_ID	New	Cm1custpart	A number representing the ID of the partition. The field have to be populated with the cycle code from the CM1_CYCLE_PARTITION_REL table
Gcmgdd APP	CM_USR_GRP_MEMBERS	Table	Update	RES_INSTANCE_ID	New	ENTSEQNO	Holds the resource instance ID of the resource. Updated through the “cm_usr_grp_members.udat“ upgrade script.
Gcmgdd APP	CM_USR_GRP_MEMBERS	Table	Update	GRP_PRM_INSTANCE_ID	New	ENTSEQNO	Holds the group parameter instance ID of the group parameter. Updated through the “cm_usr_grp_members.udat“ upgrade script.
Gcmgdd APP	CM_USR_GRP_MEMBERS	Table	Update	ADMIN_CONTEXT_KEY	New	CM1PIDKEY	Indicates if the partition key should be calculated according to the AGREEMENT_ID field or according to the GROUP_ID field. *0 – Indicates that the partition key is according to the AGREEMENT_ID field *1– Indicates that the partition key is according to the GROUP_ID field. Updated through the “cm_usr_grp_members.udat“ upgrade script.

DB Object Area & GDD BB	Object Name	Object Type	Object Change Type	Column Name	Column Change Type	Domain	Change Description
Gcmgdd APP	CM_USR_GRP_MEMBERS	Table	Update	ADMIN_CONTEXT_ID_KEY	New	CM1PIDKEY	Identifies the table partitioning key and relevant only for database performance. This field replaces the PARTIOTION_KEY field. Updated through the "cm_usr_grp_members.udat" upgrade script.
Gcmgdd APP	CM_USR_GRP_MEMBERS	Table	Update	PARTITION_KEY	Delete		Was replaced by the fields ADMIN_CONTEXT_KEY and ADMIN_CONTEXT_ID_KEY
Gcmgdd APP	CM_USR_GRP_MEMBERS	Table	Update	RESOURCE_PRM_CD	Delete		Not in use
Gcmgdd REF	CM_RESOURCE_TYPE	Table	Update	DISPLAY_TYPE	New		A new column: DISPLAY_TYPE was added. Used by the CM frontend screens. Defines the display type of the current resource type. This field is populated by the ABP core implementation team, for all the core implementation supported resource type. If a new display type needs to be added, it should be supported by the CM frontend.
Gcmgdd REF	CM_RESOURCE_TYPE	Table	Update	RANGE_IND	New	YESNOIND	Specifies whether this type represents a single resource or a range resource
Gcmgdd REF	CM_RESOURCE_TYPE	Table	Update	RM_IMPL_CLASS	New	CSCLASSNM	Specifies which Resource Management implementation is applied for this resource type

In the table CM_USR_GRP_MEMBERS, the partition has been changed. The column PARTITION_KEY was removed and the columns ADMIN_CONTEXT_KEY and ADMIN_CONTEXT_ID_KEY were added instead.

- The ADMIN_CONTEXT_KEY holds only 2 valid values (0 or 1) and these values cannot be changed.
- The ADMIN_CONTEXT_ID_KEY holds the mod(group_id,100) or mod(agreement_no, 100) depending on the ADMIN_CONTEXT_KEY value. The modulo both for the agreement_no and the group_id are now 100 by default but can be changed by other projects by changing the following properties:
  - amdocs.cm.partitionKey.cm_usr_grp_members.group_id.partition_key – for group_id
  - amdocs.cm.partitionKey.cm_usr_grp_members.agreement_no.partition_key – for agreement_no

Till this version there was only one partition which was also modulo 100 by default and it was only according to the group_id.

So, applications that changed the partition key of this table to a value different than modulo 100, should put the same value in the property amdocs.cm.partitionKey.cm_usr_grp_members.group_id.partition_key. The other property is new and they can change it according to their needs.

CM_USR_GRP_MEMBERS should be changed according to the project/application partition key value:

- ADMIN_CONTEXT_ID_KEY
- GRP_PRM_INSTANCE_ID
- RES_INSTANCE_ID

Running the CM1ENTSPREAD job

The job CM1ENTSPREAD (Entity spread) must run after upgrading the database.

This job reads the CUSTOMER and SUBSCRIBER table, accumulates the number of customers and subscribers that belong to each billing cycle and rating partition combination, and stores the results in the CM1_PARTITION_STATISTICS table

Remove APIs

None

Stop Using Deprecated APIs

The following APIs are declared as deprecated in this version and should not be used in the long term. As deprecated APIs can be removed from future versions, it is highly recommended to stop using these APIs. The Deprecated view in Javadoc HTML documentation lists all the deprecated methods.

Class Name	Method Name	Type	Alternative
amdocs.csm3g.sessions.interfaces.api. SubscriberServices	public SubscriberIdsInfo activateReservedSubscriber(CustomerIdInfo customerIdInfo, SubscriberIdInfo subscriberIdInfo, SubscriberTypeInfo subscriberTypeInfo, ExternalIdInfo externalIdInfo, SubscriberGeneralInfo subscriberGeneralInfo, NameInfo[] nameInfo, AddressInfo[] addressInfo, UnitIdInfo unitIdInfo, SrvAgrInfo pricePlan, SrvAgrInfo[] additionalServices, LogicalResourceInfo[] logicalResourceInfo, PhysicalResourceInfo[] physicalResourceInfo, ParameterInfo[] parameters, PayChannelIdInfo primaryEventDistribution, PayChannelIdInfo defaultRCPayChannelInfo, PayChannelIdInfo defaultOCPayChannelInfo, EventDistributionDetailsInfo[] eventDistributionDetailsInfo, ChargeDistributionDetailsInfo[] chargeDistributionDetailsInfo, BusinessEntityIdInfo businessEntityIdInfo, ActivityInfo activityInfo, ActivityDateInfo activityDateInfo, ActivityInfo[] niceResource)	Future Removal	public SubscriberIdsInfo activateReservedSubscriber(CustomerIdInfo customerIdInfo, SubscriberIdInfo subscriberIdInfo, SubscriberTypeInfo subscriberTypeInfo, ExternalIdInfo externalIdInfo, SubscriberGeneralInfo subscriberGeneralInfo, NameInfo[] nameInfo, AddressInfo[] addressInfo, UnitIdInfo unitIdInfo, SrvAgrInfo pricePlan, SrvAgrInfo[] additionalServices, LogicalResourceInfo[] logicalResourceInfo, PhysicalResourceInfo[] physicalResourceInfo, ParameterInfo[] parameters, PayChannelIdInfo primaryEventDistribution, PayChannelIdInfo defaultRCPayChannelInfo, PayChannelIdInfo defaultOCPayChannelInfo, EventDistributionDetailsInfo[] eventDistributionDetailsInfo, ChargeDistributionDetailsInfo[] chargeDistributionDetailsInfo, ResourceInfo[] resourceRangeList, BusinessEntityIdInfo businessEntityIdInfo, ActivityInfo activityInfo, ActivityDateInfo activityDateInfo, ActivityInfo[] niceResource)

Class Name	Method Name	Type	Alternative
amdocs.csm3g.sessions.interfaces.api. SubscriberServices	public void addOffer(SubscriberIdInfo subscriberIdInfo, SrvAgrInfo[] Services, LogicalResourceInfo[] logicalResourceInfo, PhysicalResourceInfo[] physicalResourceInfo, ParameterInfo[] parameters, ChargeDistributionDetailsInfo[] ChargeDistributionDetailsInfo, EventDistributionDetailsInfo[] EventDistributionDetailsInfo, GuidingResourceInfo[] GuidingResourceInfo, UserGroupIdInfo userGroupIdInfo, ResourceNameInfo[] resourceNameInfo, ActivityInfo activityInfo)	Future Removal	public void addOffer(SubscriberIdInfo subscriberIdInfo, SrvAgrInfo[] services, LogicalResourceInfo[] logicalResourceInfo, PhysicalResourceInfo[] physicalResourceInfo, ParameterInfo[] parameters, ChargeDistributionDetailsInfo[] chargeDistributionDetailsInfo, EventDistributionDetailsInfo[] eventDistributionDetailsInfo, GuidingResourceInfo[] guidingResourceInfo, UserGroupIdInfo userGroupIdInfo, ResourceNameInfo[] resourceNameInfo, ResourceInfo[] resourceRangeList, ActivityInfo activityInfo)
amdocs.csm3g.sessions.interfaces.api. SubscriberServices	public void changePricePlan(SubscriberIdInfo subscriberIdInfo, SrvAgrInfo NewPricePlan, LogicalResourceInfo[] logicalResourceInfo, PhysicalResourceInfo[] physicalResourceInfo, ParameterInfo[] parameters, ChargeDistributionDetailsInfo[] ChargeDistributionDetailsInfo, EventDistributionDetailsInfo[] eventDistributionDetailsInfo, GuidingResourceInfo[] guidingResourceInfo, UserGroupIdInfo userGroupIdInfo, ResourceNameInfo[] resourceNameInfo, ActivityInfo activityInfo)	Future Removal	public void changePricePlan(SubscriberIdInfo subscriberIdInfo, SrvAgrInfo newPricePlan, LogicalResourceInfo[] logicalResourceInfo, PhysicalResourceInfo[] physicalResourceInfo, ParameterInfo[] parameters, ChargeDistributionDetailsInfo[] chargeDistributionDetailsInfo, EventDistributionDetailsInfo[] eventDistributionDetailsInfo, GuidingResourceInfo[] guidingResourceInfo, UserGroupIdInfo userGroupIdInfo, ResourceNameInfo[] resourceNameInfo, ResourceInfo[] resourceRangeList, ActivityInfo activityInfo)

Class Name	Method Name	Type	Alternative
amdocs.csm3g.sessions.interfaces.api. SubscriberServices	public void changeSubscriberOffers(SubscriberIdInfo subscriberIdInfo, SrvAgrInfo[] offersToAdd, SrvAgrInfo[] offersToRemove, SrvAgrInfo[] offersToUpdateEffDate, SrvAgrInfo[] offersToUpdateExpDate, LogicalResourceInfo[] logicalResourceInfo, PhysicalResourceInfo[] physicalResourceInfo, ParameterInfo[] parameterInfo, ChargeDistributionDetailsInfo[] chargeDistributionDetailsInfo, EventDistributionDetailsInfo[] eventDistributionDetailsInfo, GuidingResourceInfo[] guidingResourceInfo, UserGroupResourceInfo[] userGroupResourceInfo, ActivityInfo activityInfo)	Future Removal	public void changeSubscriberOffers(SubscriberIdInfo subscriberIdInfo, SrvAgrInfo[] offersToAdd, SrvAgrInfo[] offersToRemove, SrvAgrInfo[] offersToUpdateEffDate, SrvAgrInfo[] offersToUpdateExpDate, LogicalResourceInfo[] logicalResourceInfo, PhysicalResourceInfo[] physicalResourceInfo, ParameterInfo[] parameterInfo, ChargeDistributionDetailsInfo[] chargeDistributionDetailsInfo, EventDistributionDetailsInfo[] eventDistributionDetailsInfo, GuidingResourceInfo[] guidingResourceInfo, UserGroupResourceInfo[] userGroupResourceInfo, ResourceInfo[] resourceRangeList, ActivityInfo activityInfo)

Class Name	Method Name	Type	Alternative
amdocs.csm3g.sessions.interfaces.api. SubscriberServices	public SubscriberIdsInfo createNewActivateSubscriber(CustomerIdInfo customerIdInfo, ExternalIdInfo subscriberExternalIdInfo, UnitIdInfo unitIdInfo, SubscriberTypeInfo subscriberTypeInfo, SubscriberGeneralInfo subscriberGeneralInfo, NameInfo[] nameInfo, AddressInfo[] addressInfo, SrvAgrInfo[] offers, LogicalResourceInfo[] logicalResourceInfo, PhysicalResourceInfo[] physicalResourceInfo, ParameterInfo[] parameterInfo, PayChannelIdInfo defaultOCPayChannel, PayChannelIdInfo defaultRCPayChannelIdInfo, PayChannelIdInfo primaryEventPayChannelIdInfo, ChargeDistributionDetailsInfo[] chargeDistributionDetailsInfo, EventDistributionDetailsInfo[] eventDistributionDetailsInfo, GuidingResourceInfo[] guidingResourceInfo, UserGroupResourceInfo[] userGroupResourceInfo, BusinessEntityIdInfo businessEntityIdInfo, ActivityInfo activityInfo, ActivityDateInfo activityDateInfo)	Future Removal	public SubscriberIdsInfo createNewActivateSubscriber(CustomerIdInfo customerIdInfo, SubscriberIdInfo predefinedSubscriberIdInfo, UnitIdInfo predefinedSubscriberUnitIdInfo, SubscriberTypeInfo subscriberTypeInfo, ExternalIdInfo externalIdInfo, SubscriberGeneralInfo subscriberGeneralInfo, NameInfo[] nameInfo, AddressInfo[] addressInfo, UnitIdInfo unitIdInfo, SrvAgrInfo pricePlan, SrvAgrInfo[] additionalServices, LogicalResourceInfo[] logicalResourceInfo, PhysicalResourceInfo[] physicalResourceInfo, ParameterInfo[] parameters, PayChannelIdInfo primaryEventDistributionInfo, PayChannelIdInfo defaultRCPayChannelInfo, PayChannelIdInfo defaultOCPayChannelInfo, EventDistributionDetailsInfo[] eventDistributionDetailsInfo, ChargeDistributionDetailsInfo[] chargeDistributionDetailsInfo, ResourceInfo[] resourceRangeList, BusinessEntityIdInfo businessEntityIdInfo, ActivityInfo activityInfo, ActivityDateInfo activityDateInfo, ActivityInfo[] niceResource)

Class Name	Method Name	Type	Alternative
amdocs.csm3g.sessions.interfaces.api. SubscriberServices	public SubscriberIdsInfo createNewActivateSubscriber(CustomerIdInfo customerIdInfo, SubscriberTypeInfo subscriberTypeInfo, ExternalIdInfo externalIdInfo, SubscriberGeneralInfo subscriberGeneralInfo, NameInfo[] nameInfo, AddressInfo[] addressInfo, UnitIdInfo unitIdInfo, SrvAgrInfo pricePlan, SrvAgrInfo[] additionalServices, LogicalResourceInfo[] logicalResourceInfo, PhysicalResourceInfo[] physicalResourceInfo, ParameterInfo[] parameters, PayChannelIdInfo primaryEventDistribution, PayChannelIdInfo defaultRCPayChannelInfo, PayChannelIdInfo defaultOCPayChannelInfo, EventDistributionDetailsInfo[] eventDistributionDetailsInfo, ChargeDistributionDetailsInfo[] chargeDistributionDetailsInfo, BusinessEntityIdInfo businessEntityIdInfo, ActivityInfo activityInfo, ActivityDateInfo activityDateInfo, ActivityInfo[] niceResource)	Future Removal	public SubscriberIdsInfo createNewActivateSubscriber(CustomerIdInfo customerIdInfo, SubscriberIdInfo predefinedSubscriberIdInfo, UnitIdInfo predefinedSubscriberUnitIdInfo, ExternalIdInfo subscriberExternalIdInfo, UnitIdInfo unitIdInfo, SubscriberTypeInfo subscriberTypeInfo, SubscriberGeneralInfo subscriberGeneralInfo, NameInfo[] nameInfo, AddressInfo[] addressInfo, SrvAgrInfo[] offers, LogicalResourceInfo[] logicalResourceInfo, PhysicalResourceInfo[] physicalResourceInfo, ParameterInfo[] parameterInfo, PayChannelIdInfo defaultOCPayChannelInfo, PayChannelIdInfo defaultRCPayChannelIdInfo, PayChannelIdInfo primaryEventPayChannelIdInfo, ChargeDistributionDetailsInfo[] chargeDistributionDetailsInfo, EventDistributionDetailsInfo[] eventDistributionDetailsInfo, GuidingResourceInfo[] guidingResourceInfo, UserGroupResourceInfo[] userGroupResourceInfo, ResourceInfo[] resourceRangeList, BusinessEntityIdInfo businessEntityIdInfo, ActivityInfo activityInfo, ActivityDateInfo activityDateInfo)

Class Name	Method Name	Type	Alternative
amdocs.csm3g.sessions.interfaces.api. AgreementServices	public AgreementIdInfo createNewAgreement(CustomerIdInfo customerIdInfo, SubscriberTypeInfo subscriberTypeInfo, UnitIdInfo unitIdInfo, AgreementGeneralInfo agreementGeneralInfo, RefreshModeInfo refreshModeInfo, ContactInfo contactInfo, SrvAgrInfo pricePlan, SrvAgrInfo[] services, ParameterInfo[] parameterInfo, ChargeDistributionDetailsInfo[] chargeDistributionDetailsInfo, ActivityInfo activityInfo)	Future Removal	createNewAgreement public NewAgreementResultInfo createNewAgreement(CustomerIdInfo customerIdInfo, AgreementIdInfo predefinedAgreementIdInfo, AgreementTypeInfo agreementTypeInfo, UnitIdInfo unitIdInfo, ExternalIdInfo externalIdInfo, AgreementGeneralInfo agreementGeneralInfo, RefreshModeInfo refreshModeInfo, ContactInfo contactInfo, SrvAgrInfo pricePlan, SrvAgrInfo[] services, ParameterInfo[] parameterInfo, ChargeDistributionDetailsInfo[] chargeDistributionDetailsInfo, ActivityInfo activityInfo)
amdocs.csm3g.sessions.interfaces.api. AgreementServices	createNewAgreement public NewAgreementResultInfo createNewAgreement(CustomerIdInfo customerIdInfo, SubscriberTypeInfo subscriberTypeInfo, UnitIdInfo unitIdInfo, ExternalIdInfo externalIdInfo, AgreementGeneralInfo agreementGeneralInfo, RefreshModeInfo refreshModeInfo, ContactInfo contactInfo, SrvAgrInfo pricePlan, SrvAgrInfo[] services, ParameterInfo[] parameterInfo, ChargeDistributionDetailsInfo[] chargeDistributionDetailsInfo, ActivityInfo activityInfo)	Future Removal	createNewAgreement public NewAgreementResultInfo createNewAgreement(CustomerIdInfo customerIdInfo, AgreementIdInfo predefinedAgreementIdInfo, AgreementTypeInfo agreementTypeInfo, UnitIdInfo unitIdInfo, ExternalIdInfo externalIdInfo, AgreementGeneralInfo agreementGeneralInfo, RefreshModeInfo refreshModeInfo, ContactInfo contactInfo, SrvAgrInfo pricePlan, SrvAgrInfo[] services, ParameterInfo[] parameterInfo, ChargeDistributionDetailsInfo[] chargeDistributionDetailsInfo, ActivityInfo activityInfo)

Class Name	Method Name	Type	Alternative
amdocs.csm3g.sessions.interfaces.api. SubscriberServices	public SubscriberIdsInfo createNewReservedSubscriber(CustomerIdInfo customerIdInfo, SubscriberTypeInfo subscriberTypeInfo, ExternalIdInfo externalIdInfo, SubscriberGeneralInfo subscriberGeneralInfo, NameInfo[] nameInfo, AddressInfo[] addressInfo, UnitIdInfo unitIdInfo, SrvAgrInfo pricePlan, SrvAgrInfo[] additionalServices, LogicalResourceInfo[] logicalResourceInfo, PhysicalResourceInfo[] physicalResourceInfo, ParameterInfo[] parameters, PayChannelIdInfo primaryEventDistribution, PayChannelIdInfo defaultRCPayChannelInfo, PayChannelIdInfo defaultOCPayChannelInfo, EventDistributionDetailsInfo[] eventDistributionDetailsInfo, ChargeDistributionDetailsInfo[] chargeDistributionDetailsInfo, BusinessEntityIdInfo businessEntityIdInfo, ActivityInfo activityInfo, ActivityDateInfo activityDateInfo)	Future Removal	public SubscriberIdsInfo createNewReservedSubscriber(CustomerIdInfo customerIdInfo, SubscriberIdInfo predefinedSubscriberIdInfo, UnitIdInfo predefinedSubscriberUnitIdInfo, SubscriberTypeInfo subscriberTypeInfo, ExternalIdInfo externalIdInfo, SubscriberGeneralInfo subscriberGeneralInfo, NameInfo[] nameInfo, AddressInfo[] addressInfo, UnitIdInfo unitIdInfo, SrvAgrInfo pricePlan, SrvAgrInfo[] additionalServices, LogicalResourceInfo[] logicalResourceInfo, PhysicalResourceInfo[] physicalResourceInfo, ParameterInfo[] parameters, PayChannelIdInfo primaryEventDistributionInfo, PayChannelIdInfo defaultRCPayChannelInfo, PayChannelIdInfo defaultOCPayChannelInfo, EventDistributionDetailsInfo[] eventDistributionDetailsInfo, ChargeDistributionDetailsInfo[] chargeDistributionDetailsInfo, ResourceInfo[] resourceRangeList, BusinessEntityIdInfo businessEntityIdInfo, ActivityInfo activityInfo, ActivityDateInfo activityDateInfo)

Class Name	Method Name	Type	Alternative
amdocs.csm3g.sessions.interfaces.api. SubscriberServices	public SubscriberIdInfo createPreActivatedSubscriber(BusinessEntityIdInfo businessEntityIdInfo, SubscriberTypeInfo subscriberTypeInfo, ExternalIdInfo externalIdInfo, SubscriberGeneralInfo subscriberGeneralInfo, SrvAgrInfo pricePlan, SrvAgrInfo[] services, ParameterInfo[] parameters, LogicalResourceInfo[] logicalResourceInfo, PhysicalResourceInfo[] physicalResourceInfo, DeviceInfo[] deviceInfo, UserGroupResourceInfo[] userGroupResourceInfo)	Future Removal	public SubscriberIdInfo createPreActivatedSubscriber(SubscriberIdInfo predefinedSubscriberIdInfo, BusinessEntityIdInfo businessEntityIdInfo, SubscriberTypeInfo subscriberTypeInfo, ExternalIdInfo externalIdInfo, SubscriberGeneralInfo subscriberGeneralInfo, SrvAgrInfo pricePlan, SrvAgrInfo[] services, ParameterInfo[] parameters, LogicalResourceInfo[] logicalResourceInfo, PhysicalResourceInfo[] physicalResourceInfo, DeviceInfo[] deviceInfo, UserGroupResourceInfo[] userGroupResourceInfo, ResourceInfo[] resourceRangeList)
amdocs.csm3g.sessions.interfaces.api. RefDataMngr	public ResultTable[] getRTActivityGroupDecodeList() throws java.rmi.RemoteException, CMException	Future Removal	public ResultTable[] getRTActivityGroupDecodeList(BusinessEntityIdInfo businessEntityIdInfo)
amdocs.csm3g.sessions.interfaces.api. RefDataMngr	public ResultTable[] getRTCsmActivityReasonDecodeList()	Future Removal	public ResultTable[] getRTCsmActivityReasonDecodeList(String activityCode)
amdocs.csm3g.sessions.interfaces.api. NewAgreementConv	public SubscriberTypeInfo getSubscriberTypeInfo()	Future Removal	public AgreementTypeInfo getAgreementTypeInfo()
amdocs.csm3g.sessions.interfaces.api. AgreementSrvAgrConv	public SubscriberTypeInfo getSubscriberTypeInfo()	Future Removal	public AgreementTypeInfo getAgreementTypeInfo()
AgreementDistributionConv	public SubscriberTypeInfo getSubscriberTypeInfo()	Future Removal	public AgreementTypeInfo getAgreementTypeInfo()
amdocs.csm3g.sessions.interfaces.api. NewSubscrConv	public void setServices(SrvAgrInfo[] srvAgrInfo)	Future Removal	public void addService(SrvAgrInfo[] srvAgrInfo)
amdocs.csm3g.sessions.interfaces.api. NewAgreementConv	public void setServices(SrvAgrInfo[] srvAgrInfo)	Future Removal	public void addService(SrvAgrInfo[] srvAgrInfo)

Class Name	Method Name	Type	Alternative
amdocs.csm3g.sessions.interfaces.api.NewAgreementConv	public SubscriberTypeInfo setSubscriberTypeInfo()	Future Removal	public AgreementTypeInfo setAgreementTypeInfo()
amdocs.csm3g.datatypes.PCOfferInfo	public java.lang.String getDiscountPlan()	Future Removal	public String [] getDiscountPlans ()
amdocs.csm3g.datatypes.AgreementGeneralInfo	public byte getHasFutureRequest()	Removal	NA any more since the future information is loaded each session
amdocs.csm3g.datatypes.RTCMResourceType	public byte getIsPackage()	Removal	NA since it does not need to recognize the RM Unified Resource
amdocs.csm3g.datatypes.RTCMResourceType	public byte getMaintainInd()	Removal	NA, not needed any more, since the not maintained resources are filtered in the Customization layer according to the required resource by the Core layer, and defined by the PC
amdocs.csm3g.datatypes.ParameterHistoryGhostInfo	public int getParameterSequenceNo()	Future Removal	Public int getParameterInstanceId()
amdocs.csm3g.datatypes.AgreementInfo	public SubscriberTypeInfo getSubscriberTypeInfo()	Future Removal	public AgreementTypeInfo getAgreementTypeInfo()
amdocs.csm3g.datatypes.PCOfferInfo	public void setDiscountPlan(java.lang.String aDiscountPlan)	Future Removal	public void setDiscountPlans(java.lang.String[] aDiscountPlans)
amdocs.csm3g.datatypes.ParameterHistoryGhostInfo	public int getParameterSequenceNo()	Future Removal	public int getParameterInstanceId()
amdocs.csm3g.datatypes.AgreementInfo	public SubscriberTypeInfo getSubscriberTypeInfo()	Future Removal	public AgreementTypeInfo getAgreementTypeInfo()
amdocs.csm3g.datatypes.AgreementHeader	public SubscriberTypeInfo getSubscriberTypeInfo()	Future Removal	getAgreementTypeInfo()
amdocs.csm3g.datatypes.PCOfferInfo	public void setDiscountPlan(java.lang.String aDiscountPlan)	Future Removal	public void setDiscountPlan(java.lang.String [] aDiscountPlan)
amdocs.csm3g.datatypes.RTCMResourceType	public byte setIsPackage()	Removal	NA since it does not need to recognize the RM Unified Resource type
amdocs.csm3g.datatypes.RTCMResourceType	public byte setMaintainInd()	Removal	NA, not needed any more, since the not maintained resources are filtered in the Customization layer according to the required resource by the Core layer, and defined by the PC

Modify Changed APIs

None

Update Operational Changes

CM1XMLOAD batch process changes

In previous versions of CM1XMLOAD, the batch was responsible for:

- Importing the Product Catalog distribution into the CM reference tables
- Sending the Product Catalog distribution, using the FilePlacer, to Clarify

Starting from this version, Clarify has added a new process that imports the distribution directly from Product Catalog. Therefore the CM1XMLOAD batch process no longer sends PC distribution to Clarify. But in order to support backward compatibility a new process was added to CM, named CM1PC2CRM.

Job	Job Name	Action	Field	Description
PC – CM extract process	CM1XMLOAD	Removed	URM_CONNECT_IND	Determine which XML to extract: * If its value is 'N' than extract will not include the URM component. * If its value is 'N' than extract will include the URM component. This version, the URM connector object, automatically deals with "No URM" mode
	CM1XMLOAD	Removed	ABP_CM_INTER	
	CM1XMLOAD	Removed	EXPORT_FILE_IND	
	CM1XMLOAD	Removed	COMPRESS_FILE_IND	
PC – CM extract process for backward compatibility	CM1PC2CRM	New	COMPRESS_FILE_IND	Indicates whether the file should be zipped.

Integrate with SONAR

Modify customization configuration files to use the SONAR standard. For more details, please refer to the SONAR user guide.

Adjust Batch Processes

None

Adjust Daemon Processes

None

Update Configuration Changes

This section details all the configuration changes that were made in the core layer that impact the customization layer. Configuration properties are maintained in properties files and other various configuration files.

Properties Files

In version 6.0 CM has been enhanced to use the generic ABP Java foundation package. This enhancement, has dictates a standard property files usage. CM properties are now divided as follows:

- CM1JF.properties contains all the properties that are used by Java Foundation regardless of the run mode (EJB or Batch)
- CM1JFBatch.properties contains all the properties that are used by Java Foundation that are specific to the Batch run mode.
- CM1JFEJB.properties contains all the properties are used by Java Foundation that are specific to the EJB run mode.
- CM1Environment.properties contains all the properties that CM needs regardless of the run mode.
- CM1BatchEnvironment.properties contains all the properties that CM needs that are specific to the Batch run mode.

Update XML API XSDs

The XSDs are reliable representatives of the APIs. In case of API change, the relevant XSD will be changed accordingly. In case a customized API exists, its XSD representation should be regenerated.

Update Published XML

Core Changes

Cycle close day data is no longer valid, due to the Flexible bill cycle support.

In published TRB transactions the CustomerBillingCycleInfo is published instead of CudtomerBillingCycleAndCloseDayInfo datatype.

The impact is that the cycle close day is no longer published.

All subscriber components should acquire the billing cycle reference data from Billing, which is the owner of the CYCLE table.

Customization Changes

In case of core XSDs that have been extended in previous versions, changes must be merged into new core XSDs (or vice versa).

Update Customization Interface

Methods Supplied to Deprecated Customization

Several methods in the amdocs.csm3g.customizationexits.utils.CustomizationUtils interface are now deprecated and should be replaced by calls to the amdocs.acmarch.customization.CommonCustomizationUtils interface, provided by the Java Foundation. This interface can be accessed via the CustomizationUtils::getCommonUtils() method.

All deprecated methods and their replacements are documented in detail in the javadoc.

Exit Points Changes

None

Adjust Implementation Core Data

Some reference tables, contains data which is mandatory for CM runtime. This type of data is called "core implementation". It could exist either as IDAT rows, or core implementation. In case customization data has been loaded into these tables, 6.0 core's data has to be reloaded. The relevant tables are:

- CSM_ACTIVITY
- CSM_ACT_RSN
- CSM_FLOW_CUSTOMIZE
- CM1_GENERIC_CODES
- ADR_NAME_DEFINITION
- CM_ENT_TRANS_RULES
- CM1_ENT_NM_ADR_REL
- CM1_SEQ_ARRAY
- CSM_GENERIC_CODES
- CSM_NM_ADDR_VALID

Extract Tool

Summary of Upgrade Activities

The following table summarizes the activities that participate in the development upgrade process. The activities are presented in the recommended order of their execution. Further details about each activity are described in subsequent sections.

#	Activity	Description	Scheduling	Automatic Process	Prerequisite for Development	Comments
1	Update Message Handling Services	Check if code changes are needed due to interface change	Pre Build	No	Yes	
2	Modify Changed Database Structures				No	
3	Stop Using Deprecated APIs	Several APIs are declared as deprecated, and may be removed in the next release. This step is optional, if it is required to stop using these APIs, and start using the alternatives	Development	No	No	Performing this step will save the need to handle the “deprecated APIs removal” step in the next version
4	Update Operational Changes			??	??	
5	Integrate with SONAR	Modify customization configuration files to use the SONAR standard	Development	No	No	
6	Adjust Batch Processes	Integration with SONAR. Adjust the modified core’s scripts Operational setup	Development	No	Yes	
7	Automatic Upgrade Scripts	Script to run in order to perform the upgrade in automatic manner	Defined by the script's context	Semi Automatic	Yes	Scripts that replace the manual upgrade activities, e.g. data migration, for XML’s, DB, operational, shell scripts etc.

Update Message Handling Services

In this version the Extract Tool is integrated with the MHS. Therefore if the MHS needs to be in the classpath of the Extract Tool, other MHS configuration files, like the message bundle and MHS initialization file need to exist in the running environment of the Extract Tool.

Modify Changed Database Structures

Database changes are described in the Extract Tool Release Notes. The automatic database upgrade process is responsible for the structure and data upgrade. The data upgrade is required when there is a need to upgrade a live environment containing data from 5.5 to 6.0.

The Tiger tool supports running the upgrade.

As database objects may be used by customizations (e.g., map customization views to core table fields), any removed or modified objects may impact the customization layer, **these changes are listed here as well as in the Release Notes.**

DB Object Area & GDD BB	Object Name	Object Type	Object Change Type	Column Name	Column Change Type	Domain	Change Description
REF get1gdd	ET1_ELEMENT	TABLE	Modified	IMPORT_ID	Added	ET1REP_SEQ	
REF get1gdd	ET1_ELEMENT	TABLE	Modified	IMPORT_ACTV	Added	ET1IMP_ACTV	
REF get1gdd	ET1_ELEMENT_TEXT	TABLE	Modified	IMPORT_ID	Added	ET1REP_SEQ	
REF get1gdd	ET1_ELEMENT_TEXT	TABLE	Modified	IMPORT_ACTV	Added	ET1IMP_ACTV	
REF get1gdd	ET1_ELEMENT_RELATTIO N	TABLE	Modified	IMPORT_ID	Added	ET1REP_SEQ	
REF get1gdd	ET1_ELEMENT_RELATTIO N	TABLE	Modified	IMPORT_ACTV	Added	ET1IMP_ACTV	
REF get1gdd	ET1_ATTRIBUTE	TABLE	Modified	IMPORT_ID	Added	ET1REP_SEQ	
REF get1gdd	ET1_ATTRIBUTE	TABLE	Modified	IMPORT_ACTV	Added	ET1IMP_ACTV	
REF get1gdd	ET1_DEPLOYMENT	TABLE	Modified	IMPORT_ID	Added	ET1REP_SEQ	
REF get1gdd	ET1_DEPLOYMENT	TABLE	Modified	IMPORT_ACTV	Added	ET1IMP_ACTV	
REF get1gdd	ET1_FIELD	TABLE	Modified	IMPORT_ID	Added	ET1REP_SEQ	

DB Object Area & GDD BB	Object Name	Object Type	Object Change Type	Column Name	Column Change Type	Domain	Change Description
REF get1gdd	ET1_FIELD	TABLE	Modified	IMPORT_ACTV	Added	ET1IMP_ACTV	
REF get1gdd	ET1_GENERIC_CODES	TABLE	Modified	IMPORT_ID	Added	ET1REP_SEQ	
REF get1gdd	ET1_GENERIC_CODES	TABLE	Modified	IMPORT_ACTV	Added	ET1IMP_ACTV	
REF get1gdd	ET1_OBJ_RESOURCE	TABLE	Modified	IMPORT_ID	Added	ET1REP_SEQ	
REF get1gdd	ET1_OBJ_RESOURCE	TABLE	Modified	IMPORT_ACTV	Added	ET1IMP_ACTV	
REF get1gdd	ET1_REMARK	TABLE	Modified	IMPORT_ID	Added	ET1REP_SEQ	
REF get1gdd	ET1_REMARK	TABLE	Modified	IMPORT_ACTV	Added	ET1IMP_ACTV	
REF get1gdd	ET1_REPOSITORY_CTRL	TABLE	Modified	IMPORT_ID	Added	ET1REP_SEQ	
REF get1gdd	ET1_REPOSITORY_CTRL	TABLE	Modified	IMPORT_ACTV	Added	ET1IMP_ACTV	
REF get1gdd	ET1_RESOURCE	TABLE	Modified	IMPORT_ID	Added	ET1REP_SEQ	
REF get1gdd	ET1_RESOURCE	TABLE	Modified	IMPORT_ID	Added	ET1REP_SEQ	
REF get1gdd	ET1_VALID_VALUE	TABLE	Modified	IMPORT_ACTV	Added	ET1IMP_ACTV	
REF get1gdd	ET1_VALID_VALUE	TABLE	Modified	IMPORT_ID	Added	ET1REP_SEQ	
REF get1gdd	ET1_XML_CONFIG	TABLE	Modified	IMPORT_ACTV	Added	ET1IMP_ACTV	
REF get1gdd	ET1_XML_CONFIG	TABLE	Modified	IMPORT_ID	Added	ET1REP_SEQ	

Stop Using Deprecated APIs

The following APIs were declared as deprecated in this version and should not be used in the long term. As deprecated APIs can be removed from future versions, it is highly recommended to stop using these APIs. The Deprecated view in the Javadoc HTML documentation lists all the deprecated methods.

The old plug-in definition and implementation of version 5.0 were marked in this version as deprecated.

Class Name	Method Name	Type	Alternative
amdocs.et.extensions. PlugableExtension	all		Amdocs.et.workflow.implementation. plugin.WFPlugableExtension.java
amdocs.et.extensions. PlugIn	All		Amdocs.et.workflow.implementation. plugin.WFPlugableExtension.java
amdocs.et.extensions. PlugOut	All		Amdocs.et.workflow.implementation. plugin.WFPlugableExtension.java
amdocs.et.extensions.plugout. PhysicalAndItermidiateFilesPlugOut	All		amdocs.et.extensions.plugout. .PhysicalAndItermidiateFiles
amdocs.et.extensions.plugout. AFIncollectSummaryPlugOut	All		amdocs.et.extensions.plugout .AFIncollectSummary
amdocs.et.extensions.plugout. PathLeftTrimmer	all		amdocs.et.extensions.plugout. PathNameLeftTrimmer

Update Operational Changes

This section describes the operational changes that the applications using the Extract Tool need to take into consideration.

In the previous version, a customization layer project requiring the use of the Extract Tool as an extract engine, needed to open a customization project of the Extract Tool.

In this version customization layer applications do not need to do so. The Extract Tool deployment files, plug-ins and resource files can reside in the application build block. Opening a customization project for the Extract Tool for extract implementation is “deprecated”.

The following procedures describe how applications can develop implementation extracts without needing to open a customization project of the Extract Tool. There are two procedures, depending on whether the application is integrated with SONAR or not.

When the application is integrated with SONAR

Since the Extract Tool is fully integrated with SONAR, the application can put its deployment files and other resource files in the application’s public/static directory. As a result SONAR will copy them at runtime to a specific location where ET can use them.

For plug-ins to be included in the path of the Extract Tool classpath, the application needs to add its plug-ins to its classpath directive (the application is able to use the directive only for specific job executions).

The application needs to define two environment variables:

- ET1_ET1_COMPONENT_USE_SONAR. This variable indicates whether the component uses SONAR to initialize the environment. The possible values are 'true' and 'false'.
- ET1_COMPONENT_NAME_IN_DOMAIN. This variable indicates the name of the component in the domain that is used for initializing the component's environment for the extract.
  For example: Audit and Control is AC, replenishment is RPLBE1. The variable value is a string.

When the application is not integrated with SONAR

In this case the application can locate the deployment files and other resource files in any directory. Nevertheless, when using this method:

- ET1_DEPLOYMENT_PATH. This variable should point to the location of the application deployment and resource files (it can include several locations separated by comas) in order to permit the Extract Tool to find the files necessary for the extract execution.
- ET1_PLUGIN_CLASSPATH. This variable should point to a root directory containing application plug-in classes or including any application plug-in jars. Once again a list of jars and/or directories can be given.

The environment variable should be defined in the ***op_<app_name>_env_sh*** *script:*

Integrate with SONAR

The Extract Tool is fully integrated with SONAR.
SONAR is used for:

- Generating the appropriate message handling related files
- Preparing the runtime environment
- Indicating how the Extract Tool will find the appropriate deployment and resource files that might be in other applications.

For details on changes related to SONAR, please refer to the 'Update Operational Changes' section.

Adjust Batch Processes

Refer to the 'Update Operational Changes' section.

Automatic Upgrade Scripts

Please refer to the Upgrade Guide (Runtime) document.

Script Location and Name	Execution Prerequisites
ABP_UPG_< application ID as in the CC > V550_V600_<Unique string>.*	

Online Charging

Summary of Upgrade Activities

The following table summarizes the activities that participate in the development upgrade process. The activities are presented in the recommended order of their execution. Further details about each activity are described in subsequent sections.

#	Activity	Description	Scheduling	Automatic Process	Prerequisite for Development	Comments
1	Regenerate Valid Values	Delete all old core domains in the customization area, add all domains VVs via FOX	Development	No	No	The old solution is applicable, but not recommended
2	Adjust Application Make Files	Adjust the customized make files to the new version	Pre Build	No	No	
3	Regenerate Domains	Regenerate all customized domains	Build	Yes	Yes	
4	Modify Changed Database Structures					
5	Update Operational Changes					
6	Adjust Daemon Processes					
7	Update Configuration Changes	In case that the PC CM Load configuration files were customized in the previous version, merge the changes with the new files, delivered with this version	Development	No	Yes, if the files were previously customized	
8	Update Customization Interface	Generate new APIs	Development	Yes	No	
9	Automatic Upgrade Scripts	Script to run in order to perform the upgrade in automatic manner	Defined by the script's context	Yes	Defined by the script's context	Scripts that replace the manual upgrade activities, e.g. data migration, for XML's, DB, operational, shell scripts etc.

Regenerate Valid Values

As part of the Offline duplicate management feature, a few Oracle tables were updated:

GN1_ART_SECTION_PARAM

PARAM_CLASS	SECTION_NAME	PARAM_NAME	PARAM_VALUE
RB_OFFLINE1	OLC_APP	NAME	OLC_APP
RB_OFFLINE1	OLC_APP	MIN_DB_CONNECTIONS	1
RB_OFFLINE1	OLC_APP	MAX_DB_CONNECTIONS	10
RB_OFFLINE1	OLC_APP	GET_CONN_TIMEOUT	5
RB_OFFLINE1	OLC_APP	DSN	${OLC_TT_RTDS_CONNECT_DIRECT}
RB_OFFLINE1	OLC_APP	DB_USER	${USER}
RB_OFFLINE1	OLC_APP	DB_TYPE	ODBC
RB_OFFLINE1	OLC_APP	DB_PASSWORD	${USER}
RB_OFFLINE1	OFFLINE_APP	NAME	OFFLINE_APP
RB_OFFLINE1	OFFLINE_APP	MIN_DB_CONNECTIONS	1
RB_OFFLINE1	OFFLINE_APP	MAX_DB_CONNECTIONS	10
RB_OFFLINE1	OFFLINE_APP	GET_CONN_TIMEOUT	5
RB_OFFLINE1	OFFLINE_APP	DSN	${OLC_ORA_INSTANCE}
RB_OFFLINE1	OFFLINE_APP	DB_USER	${OLC_ORA_USER}
RB_OFFLINE1	OFFLINE_APP	DB_TYPE	ORACLE
RB_OFFLINE1	OFFLINE_APP	DB_PASSWORD	${OLC_ORA_PASSWORD}
RB_OFFLINE1	ART_RSP_FORMATTER	TCL_FILE_NAME	${ABP_OLC_CONFIG}/rb_tcl_scripts.tcl
RB_OFFLINE1	OLCRB_ADMIN_ACC	PORT	${OLC1_RB_OFFLINE_11_ADM}
RB_OFFLINE1	OLCRB_ACC	PORT	${OLC1_RB_11_DF}
RB_OFFLINE1	ROUTER	MODE	REMOTE
RB_OFFLINE1	SERVER_PARAMETERS	SVC_CONF_FILE_NAME	chg_svc_offline.conf
RB_OFFLINE1	SERVER_PARAMETERS	ROLE	RB
RB_OFFLINE1	SERVER_PARAMETERS	CONNECTION_POOLS	OLC_REF,OLC_APP,OFFLINE_APP
RB_OFFLINE1	SERVER_PARAMETERS	ACCEPTORS	OLCRB_ACC,OLCRB_ADMIN_ACC,ART_AMC_ACC
RB_OFFLINE1	SERVER_PARAMETERS	PARTITION_ENV	${OLC_TT_RTPARTITION_ID}
RB_OFFLINE1	SERVER_PARAMETERS	MAX_PARTITION_NUM	5
SYNC_DSN1	OFFLINE_APP	NAME	OFFLINE_APP
SYNC_DSN1	OFFLINE_APP	MIN_DB_CONNECTIONS	1
SYNC_DSN1	OFFLINE_APP	MAX_DB_CONNECTIONS	10
SYNC_DSN1	OFFLINE_APP	GET_CONN_TIMEOUT	5

PARAM_CLASS	SECTION_NAME	PARAM_NAME	PARAM_VALUE
SYNC_DSN1	OFFLINE_APP	DSN	${OLC_ORA_INSTANCE}
SYNC_DSN1	OFFLINE_APP	DB_USER	${OLC_ORA_USER}
SYNC_DSN1	OFFLINE_APP	DB_PASSWORD	${OLC_ORA_PASSWORD}
SYNC_DSN1	OFFLINE_APP	DB_TYPE	ORACLE
SYNC_DSN1	SERVER_PARAMETERS	SVC_CONF_FILE_NAME	sync_svc.conf
SYNC_DSN1	SERVER_PARAMETERS	ROLE	RB
SYNC_DSN1	SERVER_PARAMETERS	CONNECTION_POOLS	OFFLINE_APP
SYNC_DSN1	SERVER_PARAMETERS	ACCEPTORS	SYNC_ACC
SYNC_DSN1	SERVER_PARAMETERS	MAX_PARTITION_NUM	5
SYNC_DSN1	SYNCLOG	DUMP_CACHE_TIMEOUT	10
SYNC_DSN1	SYNCLOG	OUTPUT	SYNC_FILE_LOG
SYNC_DSN1	SYNCLOG	ENABLED	Y
SYNC_DSN1	SYNC_FILE_LOG	TYPE	FILE
SYNC_DSN1	SYNC_FILE_LOG	MAX_SIZE	4194304
SYNC_DSN1	SYNC_FILE_LOG	FILE_NAME	sync_file.log
SYNC_DSN1	SYNC_FILE_LOG	CACHE_SIZE	0
SYNC_DSN1	OLC_APP	NAME	OLC_APP
SYNC_DSN1	OLC_APP	MIN_DB_CONNECTIONS	1
SYNC_DSN1	OLC_APP	MAX_DB_CONNECTIONS	10
SYNC_DSN1	OLC_APP	GET_CONN_TIMEOUT	5
SYNC_DSN1	OLC_APP	DSN	${OLC_TT_RTDS_CONNECT_DIRECT}
SYNC_DSN1	OLC_APP	DB_USER	${USER}
SYNC_DSN1	OLC_APP	DB_TYPE	ODBC
SYNC_DSN1	OLC_APP	DB_PASSWORD	${USER}
SYNC_DSN1	SYNC_ACC	DELTA_TIME_OUT	3
SYNC_DSN1	SYNC_ACC	DIRECTORY	${ABP_PM_ROOT}/work/SYNC
SYNC_DSN1	SYNC_ACC	MEDIATION_TYPE	SYNC
SYNC_DSN1	SYNC_ACC	TIME_OUT	20
SYNC_DSN1	SYNC_ACC	NAME	SYNC
SYNC_DSN1	SYNC_ACC	PROTOCOL	SYNC_ACC
SYNC_DSN1	ART_RSP_FORMATTER	TCL_FILE_NAME	${ABP_OLC_CONFIG}/sync_tcl_scripts.tcl
SYNC_DSN1	EVENTLOG	ENABLED	N
SYNC	SERVER_PARAMETERS	MAX_HOURS_TO_KEEP	5
FR2	CUST_REF	NAME	CUST_REF
FR2	CUST_REF	MIN_DB_CONNECTIONS	1

PARAM_CLASS	SECTION_NAME	PARAM_NAME	PARAM_VALUE
FR2	CUST_REF	MAX_DB_CONNECTIONS	10
FR2	CUST_REF	GET_CONN_TIMEOUT	5
FR2	CUST_REF	DSN	${OLC_TT_GDDS_CONNECT}
FR2	CUST_REF	DB_USER	${USER}
FR2	CUST_REF	DB_PASSWORD	${USER}
FR2	CUST_REF	DB_TYPE	ODBC
FR2	AC	ENABLED	N
FR2	AC	PROXIES	AC_PROXY_1
FR2	EVENT_FORMATTER	PROTOCOL_ID	BALANCE,ESERV,OLCP
FR2	OLCFR_OLCP_ACC	PORT	${OLC1_FR_2_OLCP}
FR2	OLCFR_OLCP_ACC	MEDIATION_TYPE	OLCP
FR2	OLCFR_OLCP_ACC	NONBLOCK	Y
FR2	OLCFR_OLCP_ACC	PROTOCOL	TCP
FR2	OLCFR_OLCP_ACC	REACTOR_THREAD_NUM	3
FR2	OLCFR_OLCP_ACC	WINDOW_SIZE	18800
FR2	OLCFR_OLCP_ACC	NAME	OLCP
FR2	OLCP	POOL_SIZE	10
FR2	DUPLICATE_MANAGEMENT	ENABLED	N
FR2	EVENT_FORMATTER	ENHANCER_CONF_FILE	${ABP_OLC_CONFIG}/enhancer.xml
FR2	ART_RSP__FORMATTER	TCL_FILE_NAME	${ABP_OLC_CONFIG}/fr_tcl_scripts.tcl
FR2	OLCFR_ESERV_ACC	PORT	${OLC1_FR_2_SRV}
FR2	OLCFR_BALANCE_ACC	PORT	${OLC1_FR_2_BAL}
FR2	OLCFR_ADMIN_ACC	PORT	${OLC1_FR_2_ADM}
FR2	ROUTER	MODE	REMOTE
FR2	SERVER_PARAMETERS	SVC_CONF_FILE_NAME	fmt_svc.conf
FR2	SERVER_PARAMETERS	ROLE	FR
FR2	SERVER_PARAMETERS	CONNECTION_POOLS	OLC_REF,CUST_REF
FR2	SERVER_PARAMETERS	ACCEPTORS	OLCFR_BALANCE_ACC,OLCFR_ESERV_ACC,OLCFR_ADMIN_ACC, OLCFR_OLCP_ACC, OLCFR_DIAMETER_ACC,ART_AMC_ACC
FR2	OLCFR_DIAMETER_ACC	WINDOW_SIZE	18800

PARAM_CLASS	SECTION_NAME	PARAM_NAME	PARAM_VALUE
FR2	OLCFR_DIAMETER_ACC	REACTOR_THREAD_NUM	1
FR2	OLCFR_DIAMETER_ACC	PROTOCOL	TCP
FR2	OLCFR_DIAMETER_ACC	NONBLOCK	Y
FR2	OLCFR_DIAMETER_ACC	MEDIATION_TYPE	DIAMETER
FR2	OLCFR_DIAMETER_ACC	PORT	${OLC1_FR_2_DIAMETER_PORT}
XLA_OLC	SERVER_PARAMETERS	PARTITION_ENV	${PARTITION_ID}
XLA_OLC	TRACE_LOG	DUMP_CACHE_TIMEOUT	10
XLA_OLC	OLC_REF	NAME	OLC_REF
XLA_OLC	OLC_REF	MIN_DB_CONNECTIONS	1
XLA_OLC	OLC_REF	MAX_DB_CONNECTIONS	10
XLA_OLC	OLC_REF	GET_CONN_TIMEOUT	5
XLA_OLC	OLC_REF	DSN	${OLC_ORA_INSTANCE}
XLA_OLC	OLC_REF	DB_USER	${OLC_ORA_USER}
XLA_OLC	OLC_REF	DB_TYPE	ORACLE
XLA_OLC	OLC_REF	DB_PASSWORD	${OLC_ORA_PASSWORD}
XLA_OLC	PROC_TRX	AC_ENABLED	N
XLA_OLC	PROC_TRX	CACHE_SIZE	0
XLA_OLC	PROC_TRX	FILE_NAME	${ABP_PM_ROOT}/work/SYNC/SYNC.data
XLA_OLC	PROC_TRX	MAX_REC_IN_FILE	200
XLA_OLC	PROC_TRX	MAX_SIZE	40000
XLA_OLC	PROC_TRX	PERSISTENT_COUNTER_FILE	data/proc_trx_file_${OLC_TT_RTDS_CONNECT_DIRECT}_count.dat
XLA_OLC	PROC_TRX	TYPE	FILE
XLA_OLC	SERVER_PARAMETERS	CONNECTION_POOLS	REF_CONN_POOL,OLC_REF
XLA_OLC	XLA_ACC	TRACKING_TABLES	RATED_EVENT,REJECTED_EVENT,OLC1_ACTION,OLC1_PROCESSED_TRX
XLA_OLC	TRACE_LOG	DUMP_CACHE_TIMEOUT	10
XLA_OLC	XLA_ACC	NAME	XLA
OLC	SESSION_SERIALIZATION	ENABLED	N
OLC	OLCFR_ADMIN_ACC	NAME	ADMIN
OLC	OLCFR_BALANCE_ACC	NAME	BALANCE
OLC	OLCFR_DIAMETER_ACC	NAME	DIAMETER

PARAM_CLASS	SECTION_NAME	PARAM_NAME	PARAM_VALUE
OLC	OLCFR_ESERV_ACC	NAME	ESERV
OLC	OLCRB_ACC	NAME	REMOTE
OLC	OLCRB_ADMIN_ACC	NAME	ADMIN
FR1	DUPLICATE_MANAGEMENT	ENABLED	N
RB11	OFFLINE_APP	NAME	OFFLINE_APP
RB11	OFFLINE_APP	MIN_DB_CONNECTIONS	1
RB11	OFFLINE_APP	MAX_DB_CONNECTIONS	10
RB11	OFFLINE_APP	GET_CONN_TIMEOUT	5
RB11	OFFLINE_APP	DSN	${OLC_ORA_INSTANCE}
RB11	OFFLINE_APP	DB_USER	${OLC_ORA_USER}
RB11	OFFLINE_APP	DB_TYPE	ORACLE
RB11	OFFLINE_APP	DB_PASSWORD	${OLC_ORA_PASSWORD}
RB12	OFFLINE_APP	NAME	OFFLINE_APP
RB12	OFFLINE_APP	MIN_DB_CONNECTIONS	1
RB12	OFFLINE_APP	MAX_DB_CONNECTIONS	10
RB12	OFFLINE_APP	GET_CONN_TIMEOUT	5
RB12	OFFLINE_APP	DSN	${OLC_ORA_INSTANCE}
RB12	OFFLINE_APP	DB_USER	${OLC_ORA_USER}
RB12	OFFLINE_APP	DB_TYPE	ORACLE
RB12	OFFLINE_APP	DB_PASSWORD	${OLC_ORA_PASSWORD}
ART	SERVER_PARAMETERS	PROCESS_NAME	ART
ART	ART_AMC_ACC	PROTOCOL	AMC
ART	ART_AMC_ACC	MEDIATION_TYPE	ADMIN_AMC
ART	ART_AMC_ACC	AMC_SLEEP	10
OLC	PERFDATALOG	ENABLED	Y
OLC	PERFORMANCE_MONITOR	ENABLED	Y
FR1	SERVER_PARAMETERS	PROCESS_NAME	FR
RB11	SERVER_PARAMETERS	PROCESS_NAME	RB
RB12	SERVER_PARAMETERS	PROCESS_NAME	RB
RB_OFFLINE1	SERVER_PARAMETERS	PROCESS_NAME	RB_OFFLINE
CDR	SERVER_PARAMETERS	PROCESS_NAME	CDR
SED_DSN1	SERVER_PARAMETERS	PROCESS_NAME	SED_DSN
SYNC_DSN1	SERVER_PARAMETERS	PROCESS_NAME	SYNC

PARAM_CLASS	SECTION_NAME	PARAM_NAME	PARAM_VALUE
EXTB	SERVER_PARAMETERS	PROCESS_NAME	EXTB
RB11	SERVER_PARAMETERS	PARTITION_ENV	${OLC_TT_RTPARTITION_ID}
RB12	SERVER_PARAMETERS	PARTITION_ENV	${OLC_TT_RTPARTITION_ID}
RB11	SERVER_PARAMETERS	MAX_PARTITION_NUM	5
RB12	SERVER_PARAMETERS	MAX_PARTITION_NUM	5
FR1	OLCFR_OLCP_ACC	NAME	OLCP
XLA_OLC_DSN1	XLA_ACC	BOOKMARK	${USER}_DSN1

GN1_ART_ERROR_CODES

ERROR_CODE	APPL_CODE	ERROR_TYPE	SEVERITY	PARAMETERS_EN
10096	OLC	SY	H	The Oracle connection is unavailable on logical destination %PARAM1%
10097	OLC	SY	H	Rating partition environment is not found
10098	OLC	SY	H	Data in oracle table OLC1_DUP_SYNCH_STATUS is not found for %PARAM1% partition key and %PARAM2% rating partition
10099	OLC	SY	H	Validity period in sync Status table is expired for rating partition: %PARAM1%
21180	OLC	DU	W	Duplicate event detected, Md ID = %PARAM1%, Session ID = %PARAM2%, Req Num = %PARAM3%, Account Ref = %PARAM4%
21190	OLC	DU	W	SYNC file: %PARAM1% is corrupted
27001	XLA	DU	W	The message type is not found in transaction record : MD ID - %PARAM1% , Session ID - %PARAM2% , Request Num - %PARAM3% , PCN - %PARAM4% , Unique Proc ID - %PARAM5%
27002	XLA	DU	W	Bulk of trasaction records contains duplicate records that already exist in OLC1_PROCESSED_TRX table
15303	OLC	GU	H	Partition_ID field in GD1_CUSTOMER_DETAILS table is empty
12001	OLC	SY	C	The AMC Event %PARAM1% AMC Proxy Initialization failure
12002	OLC	SY	C	The AMC Event %PARAM1% AMC Proxy finalization failure
12003	OLC	SY	H	The AMC Event Request Context send message %PARAM1% failure
12004	OLC	SY	C	The AMC Event %PARAM1% open failure
12005	OLC	SY	C	The AMC Event %PARAM1% AMC Proxy Initialization failure
12006	OLC	SY	C	The AMC Event %PARAM1% AMC Proxy finalization failure
12007	OLC	SY	H	The AMC Event Acceptor get message data failure

ERROR_CODE	APPL_CODE	ERROR_TYPE	SEVERITY	PARAMETERS_EN
12008	OLC	SY	C	The AMC Event Acceptor dispatch %PARAM1% failure
430	ART	SY	M	Event Handler's maximal thread number could not be less than minimal thread number
10100	OLC	RA	H	Pricing Engine is not initialized
21172	OLC	SY	W	XML Attribute value is not valid. Attribute:%PARAM1%, value is:%PARAM2%
21171	OLC	SY	W	XML Attribute types combination is unknown. Attribute path:%PARAM1%, Source type:%PARAM2%, Dest type:%PARAM3%
14111	OLC	EA	W	Incorrect file order. Counter value is bigger in Counter table for Source Id: %PARAM1%, Token Id: %PARAM2% ,Process Code: %PARAM3% and Counter Type: %PARAM4% than in file %PARAM5%
14112	OLC	EA	H	Cannot find the connection pool name of %PARAM1% Id: %PARAM1%, Token Id: %PARAM2% ,Process Code: %PARAM3% and Counter Type: %PARAM4% than in file %PARAM5%
21200	OLC	SY	W	Rollback error: %PARAM1%

GN1_ART_MD_TYPE_REF

MD_TYPE	APPLICATION_CODE
SYNC	OLC

GN1_ART_BOOT_LIBS

LIBRARY	REGISTRATOR	APPLICATION_CODE
golcdmn	registerOutputCreators	SYNC_DSN1
golcdmn	registerSyncLogHandlerInitializer	SYNC_DSN1
golcdmn	SyncAcceptorRegistrator	SYNC_DSN1
golcses	registerSessionParamInitializer	OLC
golcchg	registerProcTrxInitializer	OLC
golcrtr	registerRTRBasicRefreshedInitializer	RB11
golcrtr	registerRTRBasicRefreshedInitializer	RB12
golcrtr	registerPricingEngineInitializer	RB11
golcrtr	registerPricingEngineInitializer	RB12
gartapi	registerAMCAcceptorCreator	OLC
golcfw	registerFwBasicRefresherInitializer	XLA
golcchg	creatorsDLFunction	XLA
golcfw	registerConnPoolsInitializer	OLC
golcfw	registerConnPoolsInitializer	XLA
golcfw	creatorsDLFunction	XLA
golcfw	registerPerfDataPublisherInitializer	OLC
golcfmt	registerFmtBasicRefresherInitializer	FR2
golcrtr	registerPricingEngineInitializer	RB_OFFLINE1
golcfmt	registerFmtBasicRefresherInitializer	CDR

GN1_ART_IMPL_VER_REF

MSG_TYPE	APPLICATION_CODE	IMPL_TYPE	VERSION	LIBRARY	REGISTRATOR
104	SYNC_DSN1	12	1	golcchg	register_ProcTrxImpl
104	SYNC_DSN1	14	1	golcchg	register_ProcTrxFileImpl
105	SYNC_DSN1	13	1	golcchg	register_SyncStatusImpl
106	OLC	3	1	golcrtr	register_RaterImpl
101	XLA	0	1	golcexp	XLAProcessedTrxRegister
119	OLC	50	1	gartapi	register_AMCImpl
47	RB_OFFLINE1	12	1	golcchg	register_ProcTrxImpl
47	RB_OFFLINE1	13	1	golcchg	register_SyncStatusImpl
46	RB_OFFLINE1	12	1	golcchg	register_ProcTrxImpl
46	RB_OFFLINE1	13	1	golcchg	register_SyncStatusImpl
43	RB_OFFLINE1	12	1	golcchg	register_ProcTrxImpl
43	RB_OFFLINE1	13	1	golcchg	register_SyncStatusImpl
38	RB_OFFLINE1	12	1	golcchg	register_ProcTrxImpl
38	RB_OFFLINE1	13	1	golcchg	register_SyncStatusImpl
36	RB_OFFLINE1	12	1	golcchg	register_ProcTrxImpl
36	RB_OFFLINE1	13	1	golcchg	register_SyncStatusImpl
34	RB_OFFLINE1	12	1	golcchg	register_ProcTrxImpl
34	RB_OFFLINE1	13	1	golcchg	register_SyncStatusImpl
32	RB_OFFLINE1	12	1	golcchg	register_ProcTrxImpl
32	RB_OFFLINE1	13	1	golcchg	register_SyncStatusImpl
31	RB_OFFLINE1	12	1	golcchg	register_ProcTrxImpl
31	RB_OFFLINE1	13	1	golcchg	register_SyncStatusImpl

GN1_ART_PROCESS_CONFIG

PROC_NAME	PROC_ID	HOST	PROC_GROUP	START_SCRIPT_PARAMS	PROC_TYPE
RB_OFFLINE1	221	${HOST}	OLC	CONNECT_MODE=RATER AUTO_XML_CREATION_FLAG=Y	RB
SYNC_DSN1	301	${HOST}	OLC	CONNECT_MODE=NONE, AUTO_XML_CREATION_FLAG=N	RB
FR2	102	${HOST}	OLC	CONNECT_MODE=MAF, AUTO_XML_CREATION_FLAG=N	FR

OLC1_MSG_TYPE_PARAM_REF

MSG_TYPE	PARAM_NAME	PARAM_VALUE
3	IS_CHARGE	Y
4	IS_CHARGE	Y
30	IS_CHARGE	N
31	IS_CHARGE	Y
32	IS_CHARGE	Y

MSG_TYPE	PARAM_NAME	PARAM_VALUE
33	IS_CHARGE	N
34	IS_CHARGE	Y
35	IS_CHARGE	N
36	IS_CHARGE	Y
37	IS_CHARGE	N
38	IS_CHARGE	Y
39	IS_CHARGE	N
40	IS_CHARGE	N
41	IS_CHARGE	N
42	IS_CHARGE	N
43	IS_CHARGE	Y
44	IS_CHARGE	N
45	IS_CHARGE	N
46	IS_CHARGE	Y
47	IS_CHARGE	Y

All customized Fox objects need to be regenerated prior to performing the upgrade.

Adjust Application Make Files

The following table summarizes the customized Make file changes that are relevant for version 6.0.

Item	Change Type	Description	Required Steps
Oracle PRO-C files build mode	New	When MODE=ANSI, a program complies fully with the ANSI SQL standard	Add it to your Make file.

Regenerate Domains

List of new domains which were created to support new duplicate management oracle tables: OLC1_PROCESSED_TRX and OLC1_DUP_SYNCH_STATUS.

Domain Name	Type (length)	Description
OLC1PATION	Number(5)	Rating partition identifier
OLC1PAYCHNO	Number(10)	Pay Channel identifier
OLC1PRDKEY	Number(2)	Period key
OLC1PTKEY	Number(5)	Partition key
OLC1REQNUM	Number(10)	The event counter/request number.
OLC1SESID	Number(18)	A unique identifier of the session/call managed by the mediation system.
OLC1TIMET	Number(10)	time_t.
OLC1UPROCID	Number(10)	A unique number that identifies the process system-wide.

The source area: ~ccip/bb/golcgdd/v60_0/olc/src.

The object area: ~ccip/proj/olcgdd600V64/olc_generated.

All customized domain objects need to be regenerated prior to performing the upgrade.

Modify Changed Database Structures

Database changes are described in the Online Charging Release Notes. The automatic database upgrade process is responsible for the structure and data upgrade. The data upgrade is required when there is a need to upgrade a live environment containing data from 5.5 to 6.0.

The Tiger tool supports running the upgrade.

As database objects may be used by customizations (e.g., map customization views to core table fields), and removed or modified objects may impact the customization layer, **these changes are listed here in addition to the Release Notes.**

DB Object Area & GDD BB	Object Name	Object Type	Object Change Type	Column Name	Column Change Type	Domain	Change Description
bb: golcgdd/olc proj: olcgdd600V64	olc1_session_cre.ttsql	TT table	Replace data type		Replace data type INT with INTEGER		
bb: golcgdd/olc proj: olcgdd600V64	olc1_session_cre.ttsql	TT table	Add new column	CYCLE_CODE	INTEGER		
bb: golcgdd/olc proj: olcgdd600V64	olc1_session_cre.ttsql	TT table	Add new column	EVENT_OBJECT_TYPE_ID	INTEGER		
bb: golcgdd/olc proj: olcgdd600V64	olc1_session_cre.ttsql	TT table	Add new column	CUSTOMER_ID	INTEGER		
bb: golcgdd/olc proj: olcgdd600V64	olc1_session_cre.ttsql	TT table	Add new column	UNIQUE_SESSION_ID	INTEGER		
bb: golcgdd/olc proj: olcgdd600V64	olc1_session_cre.ttsql	TT table	Add new column	ATTRIBUTE_LIST	VARBINARY(5120)		
bb: golcgdd/olc proj: olcgdd600V64	olc1_processed_trx_cre.ttsql	TT table	Add sequence				Create sequence ProcTrxSeq increment by 1. minvalue=1 maxvalue=7500 cycle
bb: golcgdd/olc proj: olcgdd600V64	olc1_processed_trx_cre.ttsql	TT table	Add new column	ATTRIBUTE_LIST	VARBINARY(5120)		

DB Object Area & GDD BB	Object Name	Object Type	Object Change Type	Column Name	Column Change Type	Domain	Change Description
bb: golcgdd/olc proj: olcgdd600V64	olc1_processed_trx_cre.ttsql	TT table	Add new column	MSG_TYPE	INTEGER		
bb: golcgdd/olc proj: olcgdd600V64	olc1_svc_reserv_ref	Oracle table	Remove the sequence, it is now an application sequence				
bb: golcgdd/olc proj: olcgdd600V64	olc1_svc_reserv_ref_cre_pk.sql	Oracle table	Add the BE field to the PK of the table.			olc1quot_BE_NN	Now, the pk is: Service_Seq and BE
bb: golcgdd/olc proj: olcgdd600V64	olc1_quota_ref_cre.sql	Oracle table	Add new column	BE	NUMBER (9) NOT NULL	olc1quot_BE_NN	
bb: golcgdd/olc proj: olcgdd600V64	olc1_quota_ref_cre_pk.sql	Oracle table	Add the BE field to the PK of the table.				Now, the pk is: Service_Seq, BE, Dimention_No and Q_Type
bb: golcgdd/olc proj: olcgdd600V64	olc1_quota_ref_cre_1fk.sql	Oracle table	A new FK with Service_Seq and BE to the OLC1_SVC_RESERVE_REF table.				
bb: golcgdd/olc proj: olcgdd600V64	olc1_processed_trx_cre.sql	Oracle table	New table				New table to support duplicate management
bb: golcgdd/olc proj: olcgdd600V64	olc1_dup_sync_status_cre.sql, olc1_dup_sync_status_cre_pk.sql	Oracle table	New table				New table to support duplicate management

Update Operational Changes

OLC ports

The following ports were added:

- PARTITION_ID
- OLC1_FR_1_DIAMETER_PORT
- OLC1_FR_2_DIAMETER_PORT
- OLC1_RB_OFFLINE_1_DF
- OLC1_RB_OFFLINE_2_DF
- OLC1_RB_OFFLINE_3_DF
- OLC1_RB_OFFLINE_4_DF
- OLC1_RB_OFFLINE_1_ADM
- OLC1_RB_OFFLINE_2_ADM
- OLC1_RB_OFFLINE_3ADM
- OLC1_RB_OFFLINE_4ADM
- OLC1_FR_2_OLCP
- OLC1_FR_2_SRV
- OLC1_FR_2_BAL
- OLC1_FR_2_ADM
- OLC1_RB_21_DF
- OLC1_RB_21_ADM
- OLC1_RB_22_DF
- OLC1_RB_22_ADM
- OLC1_RB_31_DF
- OLC1_RB_31_ADM
- OLC1_RB_32_DF
- OLC1_RB_32_ADM
- OLC1_RB_41_DF
- OLC1_RB_41_ADM
- OLC1_RB_42_DF
- OLC1_RB_42_ADM
- OLC1_SED_2_ADM
- OLC1_SED_3_ADM
- OLC1_SED_4_ADM
- OLC1_SYNC_1_ADM
- OLC1_SYNC_2_ADM
- OLC1_SYNC_3_ADM
- OLC1_SYNC_4_ADM

OLC Var area

Directory SYNC was created under $ABP_PM_ROOT/work.

Adjust Daemon Processes

- XLA Daemon – A new functionality was added to support duplicate management. XLA daemon should transfer charge records from the TimesTen table 'OLC1_PROCESSED_TRX' into the Oracle table 'OLC1_PROCESSED_TRX'. The XLA Daemon writes these records into a log file, in case Oracle is unavailable.
- Sync Daemon – A new daemon which is responsible for transferring records from files created by the XLA Daemon into the Oracle table 'OLC1_PROCESSED_TRX'. It is also responsible for updating the status information of its partition (DSN) in the Oracle table. 'OLC1_DUP_SYNC_STATUS' on periodical basis. There should be one Sync Daemon per TT DSN.
- RB offline Daemon – A new RB server which is dedicated for CDR processing (non-real-time). There should be one RB server per TT DSN.

Update Configuration Changes

This section details all configuration changes that were made in the core layer that impact the customization layer. Configuration properties are maintained in properties files and other various configuration files.

- sessionAddParams.xml – A new xml file representing the structure of the field attribute_list in the olc1_session table.
- duplicateAddParams.xml – A new xml file representing the structure of the field attribute_list in the olc1_processed_trx table.
- sync_svc.conf – A new file for SYNC daemon that represents the business flow of two transactions: update sync status table and process sync file.
- chg_svc_offline.conf – A new file for the RB_OFFLINE process that represents the business flow of all charge events that the RB processes.
- enhancer_offline.xml – A new file for CDR Processing. Contains a list of all stations in response and in request phase that need to be processed, per event.
- enhancer.dtd – Added a RESPONSE_STATION_LIST tag to support a new enhancer_offline.xml file.
- station_ValidDestinations.xml – A new request xml file for CDR Processing.
- station_OracleDestAvailability.xml – A new response xml file for CDR Processing.
- sync_tcl_scripts.tcl – A new empty tcl script to support the ARTResponse formatter.

Update Customization Interface

- sessionAddParams.xml – A new xml file representing the structure of the field attribute_list in the olc1_session table.
  For every event in the system we can hold a binary data of the original request attributes in the olc1_session table.
  In this xml file we hold, for every message type, the structure of the field attribute_list. Once a message type does not appear in the file, it means that no data should be kept in the attribute_list field.
  For every event it is possible to perform a full serialization of all attributes or a partial one.
  We hold a configuration parameter in the gn1_art_section_param table in order to enable/disable the option of serialization and de-serialization of this field. By default it is enabled.
- duplicateAddParams.xml – A new xml file representing the structure of the field attribute_list in the olc1_processed_trx table.
  For every event in the system we need to perform duplicate checks. This field enables us to hold a binary data of the original request attributes in the olc1_processed_trx table. Thus, for every duplicate event we can return as response to the client the original response.
  In this xml file we hold, for every message type, the structure of the field attribute_list. Once a message type does not appear in the file, it means that no data should be kept in the attribute_list field.
  For every event it is possible to perform a full serialization of all attributes or a partial one.
- sync_svc.conf – A new file for SYNC daemon that represents the business flow of two transactions: update sync status table and process sync file.
- chg_svc_offline.conf – A new file for the RB_OFFLINE process that represents the business flow of all charge events that the RB processes.
- enhancer_offline.xml – A new file for CDR Processing.
- enhancer.dtd – Added a RESPONSE_STATION_LIST tag to support a new enhancer_offline.xml file.
- station_ValidDestinations.xml – In this xml file we hold, for every event in the system, meaning charges events, a list of all stations that need to be processed. The file contains both REQUEST_STATION_LIST and RESPONSE_STATION_LIST.
- station_OracleDestAvailability.xml – In the Response Phase, we process the OracleDestAvailability stations. This station is responsible for indicating a destination as not valid.
- sync_tcl_scripts.tcl – A new empty tcl script to support the ARTResponse formatter.

Automatic Upgrade Scripts

No scripts are required in this section.

Product Catalog

Summary of Upgrade Activities

The following table summarizes the activities that participate in the development upgrade process. The activities are presented in the recommended order of their execution. Further details about each activity are described in subsequent sections.

#	Activity	Description	Scheduling	Automatic Process	Prerequisite for Development	Comments
1	Auxiliary objects implementation	Use the new Auxiliary interfaces, and fix all class implementation.	Development	No	Product Catalog extensibility Assembly	
2	Implementation Repository changes	Any process that parses Implementation Repository should be revised according to the new XSD.	Development	No	Implementation Repository.XSD	
3	ProductCatalog\ProductCatalog Incremental file type	Any process that parses ItemParameters in Pricing Item should be revised according to the new XSD	Development	No	Product Catalog Extract.xsd	ProductCatalog\ProductCatalogIncremental is not generated by default, this can be changed using indication on PC1_POLICY_VARS.
4	ProductCatalog\ProductCatalog Incremental file types are deprecated	Any process that parses these files should be revised, to use the new file types. Backward compatibility is supported however.	Development	No		ProductCatalog\ProdcutCatalogIncremental is not generated by default, this can be changed using indication on PC1_POLICY_VARS.
5	GRT “Charge code” conversion	The auxiliary definition is changed from “Logical versioned” to “Physical non versioned”	Development	Partial		The Auxiliary object “Charge code” will be converted automatically by PC converter. Any additional Auxiliary objects will be removed and need to be treated manually before the conversion.

Modify Changed Database Structures

Database changes are described in the Product Catalog Release Notes. The automatic database upgrade process is responsible for the structure and data upgrade. The data upgrade is required when there is a need to upgrade a live environment containing data from 5.5 to 6.0.

The Tiger tool supports running the upgrade.

As database objects may be used by customizations (e.g., map customization views to core table fields), and removed or modified objects may impact the customization layer, **these changes are listed here in addition to the Release Notes.**

DB Object Area & GDD BB	Object Name	Object Type	Object Change Type	Column Name	Column Change Type	Domain	Change Description
Ref	PC_ELEMENT	Table	Changed				
				OBJ_ID	Deleted		Delete column
				BUSINESS_ENTITY	Changed	pc1guid	Changed to VARCHAR(255)
				STATUS	Changed	pc1status	Changed from CHAR(1) to CHAR(2)
				GUID	New	pc1guid	New column
				VERTICAL_VERSION	New	pc1guid	New column
				ELEMENT_VALUE	Changed	pc1elmntval	Changed from CLOB to BLOB
				APP_CODE	New	pc1appcode	New column
Ref	PC_VERSION	Table	Changed				
				BUSINESS_ENTITY	Changed	pc1guid	Changed to VARCHAR(255)
				BASELINE_LOW_VER	New	pc1version	New column
				BASELINE_HIGH_VER	New	pc1version	New column
				GUID	New	pc1guid	New column
Ref	PC_CORRECTION	Table	Changed				
				EFFECTIVE_DATE	New	date	New column

DB Object Area & GDD BB	Object Name	Object Type	Object Change Type	Column Name	Column Change Type	Domain	Change Description
				DISTRIBUTED	New	boolind	New column
				CORRECTION_TYPE	New	pc1corrtype	New column
				BUSINESS_ENTITY	Changed	pc1guid	Changed to VARCHAR(255)
Ref	PC1_CONSOLIDATED_AUX	Table	New				
				NAME	New	pc1shortdes	New column
				VERSION	New	pc1version	New column
				DATA	New	pc1clob	New column
Ref	PC1_CUSTOMER_ACT	Table	New				
				CUSTOMER_ACT_SEQ	New	pc1cmactkey	New column
				ACTIVITY_ID	New	pc1cmactid	New column
				ACTIVITY_ORIGIN	New	pc1actorig	New column
				ACT_DESCRIPTION	New	pc1actdesc	New column
				REASON_CODE	New	pc1rsoncode	New column
				REASION_DESCRIPTION	New	pc1resondesc	New column
				CHARGE_CODE	New	pc1chrgcd	New column
				OC_EVENT_TYPE	New	pc1eventtp	New column
				SERVICE_FILTER	New	pc1srvcfltr	New column
				RC_INDICATION	New	pc1rcind	New column
				OC_INDICATION	New	pc1ocind	New column
				RC_UPDATE_DATE_IND	New	pc1rcdtind	New column
Ref	PC1_CUSTOMER_OFFER_ACT	Table	New				
				CUSTOMER_ACT_SEQ	New	pc1cmactkey	New column
				OFFER_ACTIVITY_ID	New	pc1ofractid	New column
				SEND_WITH_CA_IND	New	pc1sndevnt	New column

DB Object Area & GDD BB	Object Name	Object Type	Object Change Type	Column Name	Column Change Type	Domain	Change Description
				SEND_OFFER_OC_IND	New	pc1sndofroc	New column
Ref	PC1_OFFER_ACT	Table	New				
				OFFER_ACTIVITY_ID	New	pc1ofractid	New column
				ACTIVITY_NAME	New	pc1act_name	New column
				SEND_WITH_RC_IND	New	pc1sndrcind	New column
				SERVICE_FILTER	New	pc1srvcfltr	New column
				OC_EVENT_TYPE	New	pc1eventtp	New column
Ref	PC1_EXTERNAL_UPD	Table	New				
				FILE_NAME	New	pc1filename	New column
				CORRECTION	New	pc1seqno	New column
				UPDATE_DATE	New	date	New column
				STATUS	New	pc1updstat	New column
				USER_NAME	New	pc1shortdesc	New column
				EXECUTION_LOG	New	pc1clob1	New column
Ref	PC1_VERTIACLA_VERSION	Table	New				
				ID	New	pc1seqno	New column
				VERTICAL_VERSION_NAME	New	pc1ver_name	New column
				BASE_MAJOR_VERSON	New	pc1version	New column
				EFFECTIVE_DATE	New	date	New column
				EXPIRATON_DATE	New	date	New column
				EFFECTIVE_DATE_CHANGEABLE	New	pc1efg	New column
Ref	PERIOD	Table	Changed				
				FROM_HOUR	Deleted		Deleted column
				TO_HOUR	Deleted		Deleted column
				START_TIME	New	date	New column

DB Object Area & GDD BB	Object Name	Object Type	Object Change Type	Column Name	Column Change Type	Domain	Change Description
				END_TIME	New	date	New column
Ref	PC1_CORRECTION_LOG	Table	New				
				ID	New	pc1seqno	New column
				NAME	New	pc1shortdes	New column
				CORRECTION_TYPE	New	pc1corrtype	New column
				CHANGE_TYPE	New	pc1change	New column
				STATUS	New	pc1corrstat	New column
				UPD_USER	New	pc1shortdes	New column
				UPD_DATE	New	datetime	New column
Ref	PC1_VERSION_LOG	Table	New				
				ID	New	pc1version	New column
				GUID	New	pc1guid	New column
				STATUS	New	pc1verstat	New column
				UPD_USER	New	pc1shortdes	New column
				UPD_DATE	New	datetime	New column
Ref	PC1_ELEMENT_LOG	Table	New				
				ID	New	pc1seqno	New column
				GUID	New	pc1guid	New column
				NAME	New	pc1shortdes	New column
				ELEMENT_VALUE	New	pc1elmntval	New column
				UPD_USER	New	pc1shortdes	New column
				UPD_DATE	New	datetime	New column
				APP_CODE	New	pc1appcode	New column
Ref	PC1_POLICY_VARS	Table	New				
				VAR_NAME	New	pc1confprm	New column

DB Object Area & GDD BB	Object Name	Object Type	Object Change Type	Column Name	Column Change Type	Domain	Change Description
				EFFECTIVE_DATE	New	date	New column
				VAR_VALUE	New	pc1prmval	New column
				EXPIRATION_DATE	New	datetime	New column
				UPD_USER	New	pc1shortdes	New column
Ref	PC1_XML_DISTRIB	Table	New				
				RELEASE_NAME		pc1rlsname	New column
				RELEASE_SEQ_NO		pc1rlsseqno	New column
				EFFECTIVE_VER_DATE		date	New column
				VERSION		pc1version	New column
				SYS_CREATION_DATE		datetime	New column
				SYS_UPDATE_DATE		datetime	New column
				OPERATOR_ID		pc1op_id	New column
				APPLICATION_ID		Pc1app_id	New column
				DL_SERVICE_CODE		Dlsrvcd	New column
				DL_UPDATE_STAMP		dlupdstmp	New column
				RELEASE_DESC		pc1rlsname	New column
				RELEASE_TIME		datetime	New column
				RELEASE_GROUP		pc1rlsgrp	New column
				RELEASE_STAUS		pc1rlsstat	New column
				RELEASE_STS_UPDATE		datetime	New column
				RELEASE_TABLES		pc1rlstbls	New column
				XML_STORAGE_TYPE		pc1xmlstype	New column
				FILE_TYPE		pc1filetype	New column
				UNCOMPRESSED_SIZE		pc1uncomsiz	New column
				XML_LOB		pc1xml_blob	New column

DB Object Area & GDD BB	Object Name	Object Type	Object Change Type	Column Name	Column Change Type	Domain	Change Description
Ref	PC1_CHARGE_CODE	Table	New				
				CHARGE_CODE_SEQ		pc1index	New column
				CHARGE_CODE		pc1chrgcode	New column
				DESCRIPTION		pc1longdesc	New column
				CHARGE_ENTITY		pc1chrgent	New column
				REVENUE_TYPE		pc1crgrevtp	New column

Update XSD

Implementation Repository

Implementation Repository contains several changes.
The “PC Extract Schema” document contains full details of the XSD.
The following list contains only changed elements:

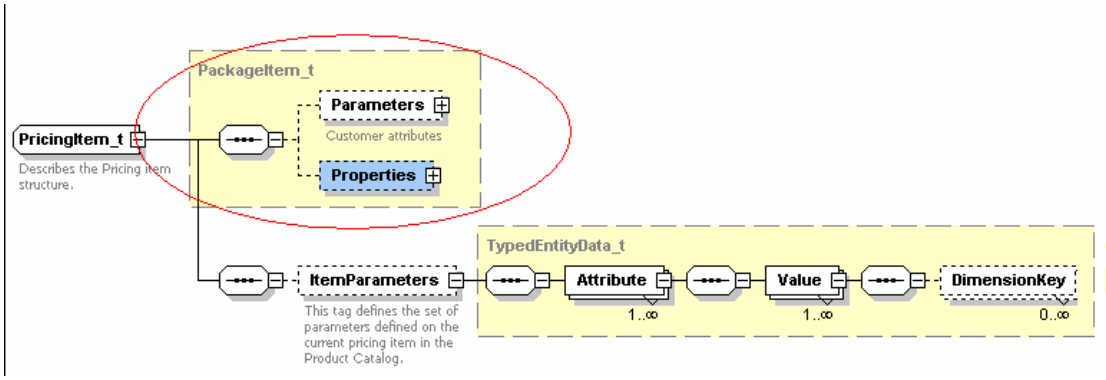
1. ProperyType
   a. The attribute entityType is replaced with the tag <EntityTypes>
   b. Support complex value
   c. Support default value.
2. Customer – Customer now contains two entities (Global and Offer level) instead of one.
3. ET Valid Values – The defaultValue element is removed, and an attribute with the same name was added.
4. ET UOM – The baseUOM, defaultUOM elements were removed, and attributes with the same names were added.
5. New Loop statement.
6. Global Rule – New Catalog Element type.
7. Service Item Type
8. Named Expression …
9. Auxiliary change position (not under extension library anymore)

New Distribution File Types

1. Period
2. Special day
3. Scale
4. Pricing Package Catalog.
5. Service Package Catalog.
6. Discount Package Catalog.
7. Offer Catalog.
8. Proposal Catalog.
9. Trailer

Product Catalog

Pricing Item – structure changes



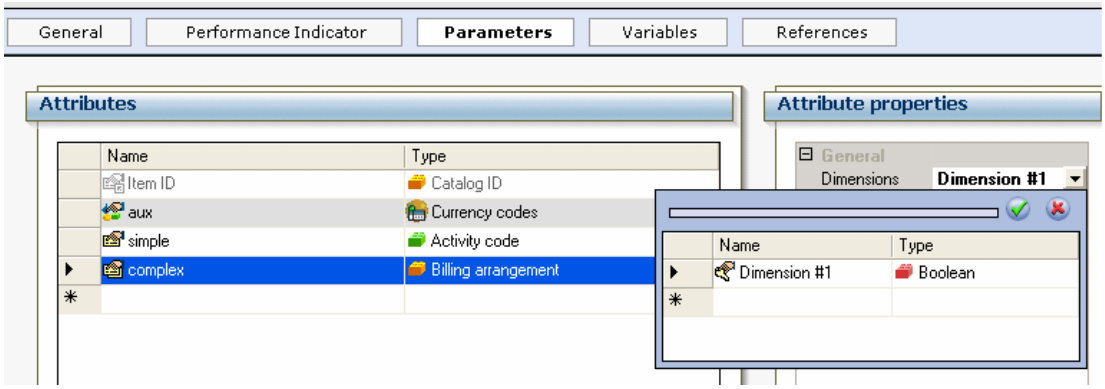
The ItemParameters tag XSD has been changed.

1. Attribute tag
   Contains the following XML attributes
   a. name – the pricing item attribute name
   b. basicType – the basic elementary type (Boolean, String, Numeric or Date)
   c. elementaryType – the elementary type name
2. Value tag
   Each Attribute contains one or more Values.
   Simple attribute will contain one value, Complex attribute may contain one or more values.
   The Value tag contains the following XML attribute:
   a. value – the value of the attribute

3. DimensionKey tag
   Each Value may contain one of more DimensionKey tags.
   Only Complex attribute will have DimensionKey list.
   DimensionKey tag contains the following XML attributes:
   a. name – value name.
   b. value – value value.
   c. dimensionName – dimension name.

The Parameters and Properties (marked in red) are not changed.

In the following example we have four attributes



4. “Item ID” and “simple” are simple attributes
5. “aux” is an Auxiliary object attribute based on the Currency codes Auxiliary type.
6. “simple” is a simple attribute based on the Activity code elementary type

7. "complex" is a complex attributed with one dimension based on "Boolean"

The following file is the XML generated for the above example:

```
<PricingItem id="159" guid="ea6e7888-5721-4481-81be-1e058f44b524"
version="000000000"name="Pricing item type #1 #1" revenueType="UC" role="Rate"
pricingItemType="Pricing item type #1">
                    <!--No parameters exist-->
                    <!--No properties exist-->
        <ItemParameters>
                <Attribute name="Item ID" basicType="Numeric" elementaryType="Catalog ID">
                        <Value value="159"/>
                </Attribute>
                <Attribute name="aux" basicType="String" elementaryType="String"
                                                        auxiliaryType="Currency codes">
                    <Value value="Currency codes"/>
                </Attribute>
                <Attribute name="simple" basicType="String" elementaryType="Activity code">
                    <Value value="CM_RERATE"/>
                </Attribute>
                <Attribute name="complex" basicType="Numeric"
                                            elementaryType="Billing arrangement">
                        <Value value="2">
                                <DimensionKey dimensionName="Dimension #1" value="1" name="Yes"/>
                        </Value>
                        <Value value="5">
                                <DimensionKey dimensionName="Dimension #1" value="0" name="No"/>
                        </Value>
                </Attribute>
        </ItemParameters>
</PricingItem>
```

Deprecated XSD

ProductCatalog and *ProductCatalogIncremental* are deprecated file types in the distribution (*ProductCatalogExtract.xsd* defines the files structure). By default PC won't generate these file types in the distribution process. However to support backward compatibility for OMSE 12.5 or other customization processes (like PC2PP) the default behavior can be override, using the variable "VERSION 5.5 Compatibility mode" defined in PC1_POILICY_VAR.

Assumptions and Limitations

There are some limitations for using these files types as described in the following section.

1. Vertical versioning and backwards compatibility cannot live together in peace. Accounts that wish to have the old ProductCatalog file type will not be able to enjoy the benefits of vertical versioning and will actually continue to work with the old versioning mechanism.

2. When the VERSION 5.5 Compatibility mode variable is set to true:
   a. PC will not allow creating vertical corrections.
   b. Finalization of horizontal correction will verify that the correction effective date is larger than the last horizontal version EVEN if the correction has only vertical elements.
   c. Release version will force all the corrections to have the same effective date EVEN if the corrections have only vertical elements.
   d. PC will continue to extract incremental XMLs at distribute time (ProductCatalogIncremental file type) and create a full XML (ProductCatalog file type) using the RTS callback.
   e. Locking vertical elements will be allowed in an horizontal fix mode, i.e. fix horizontal version may include vertical elements. The lock will either create a fix vertical version or a new vertical version effective to the fix date.
3. The ProductCatalog file type will not support the following new functionality:
   a. Customer offer level parameters.
   b. Offer new attributes (rules, attachments, ordering info).
   c. Proposals.
   d. Multi dimension properties.
   e. Equipment packages.

Update Customization Interface

Customization Public Objects

Product Catalog Extensibility interfaces that were changed:

1. IAuxiliaryHost
   a. Added method IQueryResults Query (IAuxiQuaryParams).
   b. OleDbConnection replaced with IDbConnection.
   c. OleDbTransaction replaced with IDbTransaction.
   d. Added property IDbProviderFactory ProviderFactory {get}.
2. IDbProviderFactory new interface.
3. IQueryResults new interface.
4. IAuxQuaryParams new interface.
5. QueryStatus new enum.
6. IIndexedValidValue – return type of the Index property was changed from uint to string.

Implementation Updates

This section describes the changes that are related to the Product Catalog implementation team.

Elementary Types Changes

1. The following “Core” Elementary types were changed to be “Extensible core”. This means that the implementation can add valid values to these Elementary types (but cannot delete or change existing core values):
   - Activity source
   - Notification method
   - Charge code
   - Charge origin
   - Offer actions
   - Payment frequency
   - Service filter
   - Special day types
   - Product type
2. Elementary type “Payment category” has additional valid value. Name is “Both”. Value is “BOTH".
3. Elementary type “Activity source” valid values were updated according to the following table:

Name	Value	Status
Migrate postpaid to prepaid	17	Changed
Migrate prepaid to postpaid	18	Changed
Cancel subscriber	10	Changed
Add / remove service	45	Changed
Change price plan	13	Changed
Move subscriber	19	Changed
Activate subscriber	14	Changed
Change event distribution	42	Changed
Change subscriber parameters	48	Changed
Change agreement's price plan	15	New
Service change deployed from the agreement	36	New
Agreement service change	46	New
Change price plan deployed from the agreement	51	New
Change subscriber's ownership - Cancel old subscriber	58	New
Change subscriber's ownership – Activate new subscriber	59	New
Renew subscriber	60	New

Property Types Changes

1. Property type “Should be displayed”:
   a. Has a value provider based on a new Elementary type “Should be displayed” (value provider was empty).
      This means that any property value should be “Y” or “N” or “O”.
      Otherwise a validation error will be issued.
   b. Is defined for both the Customer global and Customer offer level entities (was defined for Customer global only).
2. Property type “Parameter type” is defined for both the Customer global and Customer offer level entities (was defined for Customer global only).
3. Property type “Populate level”
   a. Has a value provider based on Elementary type “Populate level” (contains valid values “CM” and “PC”)
   b. It is no longer defined as mandatory.
   c. Is defined for both Customer global and Customer offer level entities (was defined for Customer global only).
      In all attributes occurrences, if the property has an empty value, it will be replaced with “PC”.
4. Property type “Is equipment” is no longer defined as mandatory.
5. Property type “Is mandatory” is no longer defined as mandatory.
6. Property type “Is resource” is no longer defined as mandatory.
7. Property type “Is transient” is no longer defined as mandatory.
8. Property type “Maximum offer duplicates allowed" default value changed to “0”.
9. Property type "Should be deployed" default value changed to “Yes”.
10. Property type "Product type" default value changed to “U”.

GRT Changes

1. CM Activity auxiliary is deprecated.
   Note that its physical table (PC1_CM_ACTIVITY) is not synchronized with CM tables by RTS.
   In order to continue working with the old implementation please refer to Appendix D

2. A new GRT Customer Activities based on the physical table PC1_CUSTOMER_ACT.
   The purpose of this table is to map customer activity group and reason code to a charge code.
   This GRT is synchronized with 3 CM tables (using 3 RTS callbacks):
   a. CM1_GENERIC_CODES
   b. CSM_ACT_RSN
   c. CSM_ACTIVITY

   There are several assumptions regarding the synchronization:
   d. The CM tables are synchronized, i.e. each activity reason has a corresponding activity code and description.
   e. When updating an activity code it means creating a new code, rather then changing the existing code.
   f. A combination of reason code and activity code is unique.
   g. When inserting to the 'PC1_CUSTOMER_ACT' table, the not null columns receive default values, as follows:
      i. CHARGE_CODE = 'DUMMY'
      ii. RC_INDICATION = 'N' (false)
      iii. OC_INDICATION = 'N' (false)
      iv. ACTIVITY_ORIGIN = 'CM'
3. GRT "Charge codes"
   This Auxiliary was changed to be a physical non versioned GRT. It is based on the table "PC1_CHARGE_CODE".
   The Auxiliary data will be converted automatically with the following exceptions:
   a. The auxiliary object that will be converted is the default "Charge code" object. Since the type was defined as "Multiply object", other auxiliary object may exist. These auxiliary objects will be removed. Their data can be copied before the conversion, to the "Charge code" auxiliary object.
   b. The data from "Charge code" object is copied from the last version. In case auxiliary object values were deleted in that last version, and there are references to these values in older versions, those versions will became invalid since the new GRT is non versioned and contains only the last version object values.
      To solve this problem all previous versions must be validated and fixed.

Event Changes

1. Added “Event Variable” entity under Event Item.
2. Updated Access PI indicator in Event handlers.
3. Event “Event” is changed from “Core” to “ExtensibleCore”.
   a. Attribute “Network start time” is changed from “Core” to “ExtensibleCore”.
   b. Attribute “Service filter is changed from “Core” to “ExtensibleCore”.
   c. The following attributes are defined as “Fielded”: ”Cycle month”, Cycle year”, “Cycle code”, “Event type ID”, Subscriber ID”, “Customer ID”, “Start time”, “Target cycle code”, “Source ID”.
4. Event “Nu-Rater base event” is changed from “Implementation” to “ExtensibleCore”.
   a. Attribute “Event type name” is changed from “Core” to “ExtensibleCore”.
   b. Attribute “Service receiver type” is changed from “Core” to “ExtensibleCore”.
5. Event “OC Activity event”
   a. Changed from “Implementation” to “ExtensibleCore”.
   b. Added “ExtensibleCore” attribute “Activity origin”.
   c. Attribute “Activity code” is changed from “Core” to “ExtensibleCore”.
   d. Attribute “Activity reason code” is changed from “Core” to “ExtensibleCore”.

When Event is defined as ExtensibleCore, the user can change the attribute persistence definition (the attribute must be ExtensibleCore as well).

Customer changes

Customer is split into two entities: Global and Offer level.

Customer Offers

1. Change all attribute to be static.
2. Added complex attribute “Effective activity code”.

Extension Function Changes

1. Extension function “Get period name by period value” – return type was changed to “String”.
2. Extension function “Does belong to period” – period name parameter type changed from numeric to string.
3. Extension function “Does belong to period considering special days” – period name parameter type changed from numeric to string.
4. Access PI indicator is set on all core Extension Functions.
5. The following entity methods are defined with dynamic parameters: ”Get proration factor”, Break quantity to steps", "Break quantity to steps according to accumulation", "Break quantity to steps by dimension according to accumulation", "Map segments to attribute steps", "Map segments to steps", "Map segments to steps according to accumulation".
6. The following elementary type methods were defined with dynamic parameters: "Get distance to scale leg end", "Get scale leg", "Get scale leg value".

PIT Changes

1. Attribute that belongs to the PIT’s entities, will get partition indicator set to true.
2. Updated Access PI indicator in PIT handlers.

Offer Changes

Maximum date value supported by the Rater is 31\12\2020.
Any offer that contains effective or expiration date greater then the maximum threshold will be reset to the maximum allowed.

Vertical Versioning

General

Version 6.0 introduces a new versioning mechanism – vertical versioning. The following existing elements in version 5.5 will be converted to support the new model:

- Offer
- Pricing Package
- External Package
- Discount Package
- Pricing Item

For example:

Consider the following scenario in version 5.5:

Table 1: Version history in 5.5

Action	Version	Effective date	Expiration date
New Offer	000.001.000	10 Jul 04	20 Aug 04
New Pricing Package	000.002.000	20 Aug 04	23 Sep 04
Change Offer	000.002.000	20 Aug 04	
Delete Pricing Package	000.003.000	23 Sep 04	

The converter will create two vertical versions for the Offer and for the Pricing Package.

Table 2: Vertical Version history in 6.0

Action	Vertical Version	Effective date	Expiration date
New Offer	000.000	10 Jul 04	20 Aug 04
New Pricing Package	000.000	20 Aug 04	23 Sep 04
Change Offer	000.001	20 Aug 04	
Delete Pricing Package	000.001	23 Sep 04	

Pricing Item

Pricing Item is no longer a Catalog Element.
It is an integral part of the Pricing Package, and cannot be locked as it was in version 5.5

Service Management Upgrade

PC version 5.5 had “External Package” entities imported from the SM component. These packages contains Items (Provisioning, Barring, EGI, Billing). Each Item contains parameters and properties.

Converting SM Items

Each Item contained in an “External package” will be replaced with a new model object: Service Item Type (SIT) and a Service Item based on the new Service Item Type.
The Service Item Type will be based on one of the following core SIT roles:

- Provisioning base
- Barring base
- Event group base
- Billing

The base is determined according to the old Item type.

The Service Item Type and Service Item will contain the parameters and properties of the old Item.

Not all the properties and parameters in the old mode are supported. The exact mapping between the SM model and the PC model is described in the “Merging SM to PC Implementation Guide”.

Converting "External package"

Each "External package" will be converted to a Service Package.
The type of the new package is identical to the old External Package type.
Each Service Package will contain the converted Service Item described in the previous section.
The Service Package ID will be the same as the old External Package ID.

The reference from the Offer is kept as well. For example if Offer #1 had reference to External Package "Provisioning #4", then after the conversion the External Package #4 will be converted to Service Package "Provisioning #4" and the reference from the Offer is still valid.

Provisioning

Summary of Upgrade Activities

The following table summarizes the activities that participate in the development upgrade process. The activities are presented in the recommended order of their execution. Further details about each activity are described in subsequent sections.

Activity	Description	Scheduling	Automatic Process	Prerequisite for Development	Comments
Modify Changed Database Structures					
Update Operational Changes					

Modify Changed Database Structures

Database changes are described in the Provisioning Release Notes. The automatic database upgrade process is responsible for the structure and data upgrade. The data upgrade is required when there is a need to upgrade a live environment containing data from 5.5 to 6.0.

The Tiger tool supports running the upgrade.

As database objects may be used by customizations (e.g., map customization views to core table fields), and removed or modified objects may impact the customization layer, **these changes are listed here in addition to the Release Notes.**

DB Object Area & GDD BB	Object Name	Object Type	Object Change Type	Column Name	Column Change Type	Domain	Change Description
gpvgdd/ ref	FLDS_DEF		Modified	SOURCE_NAME	Modified	FLDNM	Length increase - To store UTF8 String
gpvgdd/ ref	FLDS_DEF		Modified	PARAMTER_NAME	Modified	FLDNM	Length increase - To store UTF8 String

DB Object Area & GDD BB	Object Name	Object Type	Object Change Type	Column Name	Column Change Type	Domain	Change Description
gpvgdd/ref	DECISION_FLDS	TABLE	Update	GROUP_NAME	Modified	FLDNM	Length increase - To store UTF8 String
gpvgdd/ref	decision_rslt	TABLE	Update	TRX_NAME	Modified	FLDNM	Length increase - To store UTF8 String
gpvgdd/ref	decision_tbl	TABLE	Update	GROUP_NAME	Modified	FLDNM	Length increase - To store UTF8 String
gpvgdd/ref	dev_capacity	TABLE	Update	DVC_TP	Modified	FLDNM	Length increase - To store UTF8 String
gpvgdd/ref	dev_commands	TABLE	Update	TRX_NAME	Modified	FLDNM	Length increase - To store UTF8 String
gpvgdd/ref	dev_commands	TABLE	Update	DVC_TRX_NM_CD	Modified	FLDNM	Length increase - To store UTF8 String
gpvgdd/ref	dev_commands	TABLE	Update	LOGICAL_DVC_TP	Modified	FLDNM	Length increase - To store UTF8 String
gpvgdd/ref	dvc_not_supp_ftr	TABLE	Update	DVC_TP	Modified	FLDNM	Length increase - To store UTF8 String
gpvgdd/ref	dvc_not_supp_ftr	TABLE	Update	FLD_NAME	Modified	FLDNM	Length increase - To store UTF8 String
gpvgdd/ref	dvc_target_lvl	TABLE	Update	DVC_TP	Modified	FLDNM	Length increase - To store UTF8 String
gpvgdd/app	dvc_trx_repos	TABLE	Update	DVC_TRX_NM_CD	Modified	FLDNM	Length increase - To store UTF8 String
gpvgdd/ref	gen_key_rslt	TABLE	Update	GEN_SUBJECT	Modified	FLDNM	Length increase - To store UTF8 String
gpvgdd/ref	gen_subj_flds	TABLE	Update	GEN_SUBJECT	Modified	FLDNM	Length increase - To store UTF8 String
gpvgdd/ref	gen_subj_flds	TABLE	Update	FLD_NAME	Modified	FLDNM	Length increase - To store UTF8 String
gpvgdd/ref	grp_to_dev	TABLE	Update	TRX_NAME	Modified	FLDNM	Length increase - To store UTF8 String

DB Object Area & GDD BB	Object Name	Object Type	Object Change Type	Column Name	Column Change Type	Domain	Change Description
gpvgdd/ ref	grps_def	TABLE	Update	GROUP_NAME	Modified	FLDNM	Length increase - To store UTF8 String
gpvgdd/ ref	logic_devs	TABLE	Update	DVC_TP	Modified	FLDNM	Length increase - To store UTF8 String
gpvgdd/ ref	logic_phd	TABLE	Update	DVC_TP	Modified	FLDNM	Length increase - To store UTF8 String
gpvgdd/ app	q3	TABLE	Update	DVC_TRX_NM_ CD	Modified	FLDNM	Length increase - To store UTF8 String
gpvgdd/ app	srv_trx_rep os	TABLE	Update	SRV_TRX_NM_ CD	Modified	FLDNM	Length increase - To store UTF8 String
gpvgdd/ ref	subs_dev	TABLE	Update	DVC_TP	Modified	FLDNM	Length increase - To store UTF8 String

Update Operational Changes

PVXMLMERGE new Job

The PVXMLMERGE job is responsible for handling the following activities

1. PC xml distribution
2. PC XML incremental distribution
3. Handling Item Types and Parameter modification due to SM-PC Merger
4. Merging all XMLs to a single PV XML

This job runs at reference level.

Job	Job Name	Action	Description
PC _PV XML Merger	PVXMLMERGE	New	Job handles distributions of xmls and merges into a single XML for Provisioning.

Rating

Summary of Upgrade Activities

The following table summarizes the activities that participate in the development upgrade process. The activities are presented in the recommended order of their execution. Further details about each activity are described in subsequent sections of this section.

#	Activity	Description	Scheduling	Automatic Process	Prerequisite for Development	Comments
1	Regenerate Valid Values	Delete all old core domains in the customization area, add all domains VVs via Fox.	Development	No	No	The old solution is applicable, but not recommended
2	Adjust Application Make Files	Adjust the customized make files to the new version	Pre Build	No	No	
3	Regenerate Domains	Regenerate all customized domains	Build	Yes	Yes	
4	Modify Changed Database Structures					
5	Create New Database structures	Several new tables were added to the Pricing Engine area. These tables are required for proper Pricing Engine run.	Development	No	Yes	
6	Remove APIs	Adjust any code that uses a deprecated API that has been removed in the new version	Pre build	No	Yes	Running the build process is a simple way to find these removed APIs
7	Stop Using Deprecated APIs	Several APIs are declared as deprecated, and may be removed in the next release. This step is optional, if it is required to stop using these APIs, and start using the alternatives	Development	No	No	Performing this step will save the need to handle the “deprecated APIs removal” step in the next version
8	Modify Changed APIs	Several APIs were removed from this version and cannot be used anymore. This step is mandatory and an alternative implementation is available	Development	No	Yes	Running the build process is a simple way to find these removed APIs

#	Activity	Description	Scheduling	Automatic Process	Prerequisite for Development	Comments
9	New APIs	Several new APIs are introduced and should be used for new features that replace temporary patches given in earlier versions.	Development	No	Yes	
10	Update Operational Changes					
11	Update Configuration Changes	In case that the PE is loaded through customization, using the configuration file, new attributes need to be added to the configuration file.	Development	No	Yes, if the files were previously customized	

Regenerate Valid Values

Delete all old core domains in the customization area. Add all domains VVs via Fox.

All customized Fox objects need to be regenerated prior to performing the upgrade.

Regenerate Domains

Regenerate all customized domains.

All customized domain objects need to be regenerated prior to performing the upgrade.

Modify Changed Database Structures

Database changes are described in the Rating Release Notes. The automatic database upgrade process is responsible for the structure and data upgrade. The data upgrade is required when there is a need to upgrade a live environment containing data from 5.5 to 6.0.

The Tiger tool supports running the upgrade.

As database objects may be used by customizations (e.g., map customization views to core table fields), and removed or modified objects may impact the customization layer. **These changes are listed here in addition to the Release Notes.**

DB Object Area & GDD BB	Object Name	Object Type	Object Change Type	Column Name	Column Change Type	Domain	Change Description
pmgdd600V64	PM1_CYCLE_ CHANGE	Table	New	CUSTOMER_ID	New	NUMBER(9)	New table
pmgdd600V64	PM1_CYCLE_ CHANGE	Table	New	EFFECTIVE_DATE	New	DATE	New table
pmgdd600V64	PM1_CYCLE_ CHANGE	Table	New	NEW_CYCLE_ CODE	New	NUMBER(2)	New table
Pmgdd600V64	PM1_CYCLE_ CHANGE	Table	New	OLD_CYCLE_ CODE	New	NUMBER(2)	New table
Pmgdd600V64	PM1_CYCLE_ CHANGE	Table	New	STATUS	New	CHAR(1)	New table
Pmgdd600V64	PM1_CYCLE_ CHANGE	Table	New	SYS_CREATION_ DATE	New	DATE	New table
Pmgdd600V64	PM1_CYCLE_ CHANGE	Table	New	SYS_UPDATE_ DATE	New	DATE	New table
Pmgdd600V64	PM1_CYCLE_ CHANGE	Table	New	APPLICATION_ID	New	CHAR(6)	New table
Pmgdd600V64	REJECTED_ EVENT	Table	Modify	ERROR_CODE	Field size increased	VARCHAR2(8)	Field size increased

PM1_CYCLE_CHANGE Table

A new table defined in the Pricing Engine as part of the Change Cycle functionality. When using previous solutions for Change Cycle functionality (given as part of defect #12697 patch #9081), it is required to convert the previous stored data to the new PM1_CYCLE_CHANGE table.

Remove APIs

The following deprecated objects/methods were removed from this version. In order to adjust the customized code to this change, it is required to stop using these objects and methods.

Class Name	Method Name	Type	Removal Reason	Alternative
PricingEngine	initRerate	Method removal	The functionality was moved to the regular init method	Use the PricingEngine init method (full initialization) and then call the new setCycleInfo method for each processed cycle
UsageQueryAPI	init	Method removal	Functionality was moved to Billing	Use new APIs from Billing
UsageQueryAPI	release	Method removal	Functionality was moved to Billing	Use new APIs from Billing
UsageQueryAPI	getRatedEvents	Method removal	Functionality was moved to Billing	Use new APIs from Billing
UsageQueryAPI	getRatedEventDetails	Method removal	Functionality was moved to Billing	Use new APIs from Billing
UsageQueryAPI	getPIs	Method removal	Functionality was moved to Billing	Use new APIs from Billing
UsageQueryAPI	getPIDetails	Method removal	Functionality was moved to Billing	Use new APIs from Billing
EntityCursor	getErrorInfo	Method removal	The method was removed as the logic is now done by the IO F/W which does not output the TT or Oracle errors	Log file will contain the TT or Oracle error

A simple build process easily generates a list of all areas in the customized code that need to be fixed.

Stop Using Deprecated APIs

The following APIs were declared as deprecated in this version and should not be used in the long term. As deprecated APIs can be removed from future versions, it is highly recommended to stop using these APIs.

Class Name	Method Name	Type	Alternative
Pricing Engine	initDataDictionary	Deprecated	Use the new initDataDictionary method that uses the InitParameters input structure

Modify Changed APIs

Class Name	Method Name	Type	Removal/Change Reason	Alternative
PricingEngine	prepareConnection	Method changed	The DBConnection input parameter was changed to the generic IO F/W connection object acm::common::io::IConnection	Change the use of the prepareConnection method to receive as an input a pointer to one of the following changed classes: OracleConnection, TimesTenConnection or DualConnection
PricingEngine	initImplementationRepository	Method changed	Method was changed to receive an optional input parameter, UtString& i_encoding.	The additional encoding parameter is optional and will be set to default iso8859-1 if not supplied to the method. It is required to pass the correct encoding in any case where the system works with XMLs encoded with anything other then iso8859-1.

New APIs

The following objects/methods were added to this version. In order to adjust the customized code to this change, it is required to start using these objects and methods.

Class Name	Method Name	Type	Replacement Reason	Alternative
PricingEngine	createCycleChange	New	This new method is used in order to create cycle change entities. It is part of the cycle change feature and should be used instead of the partial solution given for defect #12697 patch #9081	
PricingEngine	setCycleInfo	New	Used for setting the cycle info for the re-rate processing. This method is called after calling the init method, for setting the PE to work in the re-rate mode	
PricingEngine	setReferenceDataStorage	New	This method is used for setting the Pricing Engine reference data storage object for the current processing thread (for Multi-threaded processing only). If the input storage object is null, then the Pricing Engine automatically sets the storage to the latest existing storage available	
PricingEngine	releaseReferenceDataStorage	New	This method releases the current processing thread storage object (for Multi-threaded processing only).	

Update Operational Changes

Removed jobs

Job name	Description	Type	Reason	Alternative
PM1CHECKPOINT	CheckPoint	Delete	TimesTen utility is used instead	
PM1SPCHECKPT	Stop CheckPoint	Delete	TimesTen utility is used instead	
PM1PIEXTRACT	PI Extract	Delete	New process	PM1UEXTPI – server activation PM1PIEXT – PI Extract request
PM1EVENTEXT	Event Extract	Delete	New process	PM1UEXTEVENT – sever activation PM1EVEXT – event extract request
PM1EVENTREC	Event Extract Recovery	Delete	New process	PM1RCVCYCLEX – Cycle Mode Recovery PM1RCVCUSTOMEREXT – Customer Mode recovery
PM1NEWCYCLE	New Cycle	Delete	Published billing transaction	
PM1NEWSTATE	New state	Delete	Published billing transaction	
PM1NURATER	NU Rater	Delete	Process moved to Billing	
PM1PIEXTREC	PI Extract Recovery	Delete	New process	PM1RCVCYCLEX – Cycle Mode Recovery PM1RCVCUSTOMEREXT – Customer Mode recovery

Update Configuration Changes

This section details all configuration changes that were made in the core layer that impact the customization layer. Configuration properties are maintained in properties files and other various configuration files.

pmPricingEngineConfig.xml Changes

The Pricing Engine configuration file was added with the following new tags:

- ParsingParameters – XML parsing related attributes.
- EventTableStatus – indicates the state of the event table. Can be one of the following values: HYBRID, FIELDED and DYNAMIC.
  If the parameter is not passed as part of the configuration, the PE sets the parameter to a default value of DYNAMIC.
- RunMode – indicates the PE run mode in relation to the “PIT with no PI” functionality. The possible values for the parameter are Always, OfflineRating and OnlineRating.
  If the parameter is not passed as part of the configuration the PE will activate all the handlers regardless of the run mode.

The Pricing Engine configuration file was added with the following new attributes:

- PartitionID – indicated the partition on which the PE instance is running. Each PE instance can be connected/run on one partition at a time. The PartitionID should be provided to the PE as part of the configuration file. If the parameter is not passed as part of the configuration the PE assumes that only one partition exists. The new PartitionID attribute location in the configuration file is under the EventProcessingParameters tag.
- MaxLoopIterations – indicates the maximum number of loops, on complex attributes, that the PE instance can perform. If the parameter is not passed as part of the configuration the PE sets the maximum to a default of 20 iterations. The new MaxLoopIterations attribute location in the configuration file is under the EventProcessingParameters tag.
- encoding – indicates the encoding the PE will use for parsing the XML files (Implementation Repository). If the parameter is not passed as part of the configuration, the PE sets the attribute to a default value of iso8859-1. The new encoding attribute location in the configuration file is under a new tag: ParsingParameters.
- MaxPrefetchRows – indicates the maximum number of rows that will be fetched at once from the DB (one round trip). This attribute sets the maximum number of rows that will be fetched form the PM1_CUSTOMER_OFFER_PARAMS table. If the parameter is not passed as part of the configuration, the PE sets the attribute to a default value of 20. The maximum configurable limitation is 128. If the parameter is set to any number greater then 128, the PE will automatically set it to the maximum of 128.

Pm1RaterProcessesConfig.xml Changes

Few configuration sections have been removed since the processes which are based on the ART platform now use specific parameters tables in the DB. The sections are:

- Usage Extract (moved to ART)
- CM2RaterExtract (old process)
- CMGenericExtract (moved to ART)
- NURater (process is part of Billing)
- PostpaidRater (moved to ART)
- PostpaidRaterRecovery (moved to ART)
- LockRater (Not used)

For more information regarding parameters configuration for rater processes refer to the Rating Business Parameter Configuration Guide (#185844)

RPR1_XML_CONFIG Table

The Extract configuration entry has been changed (object name pm1CMGenericExtractConfigurationFile.xml).

Pricing Engine

Summary of Upgrade Activities

The following table summarizes the activities that participate in the development upgrade process. The activities are presented in the recommended order of their execution. Further details about each activity are described in subsequent sections.

#	Activity	Description	Scheduling	Automatic Process	Prerequisite for Development	Comments
1	Regenerate Valid Values	Delete all old core domains in the customization area, add all domains Valid Values via Fox	Development	No	No	The old solution is applicable, but not recommended
2	Regenerate Domains	Regenerate all customized domains	Build	Yes	Yes	
3	Modify Changed Database Structures	Several table structures in the Pricing Engine area were modified / added. These tables are required for proper Pricing Engine run.	Development	No	Yes	
4	Create New Database structures	Several new tables were added to the Pricing Engine area. These tables are required for proper Pricing Engine run.	Development	No	Yes	
5	Remove APIs	Adjust any code that uses a deprecated/unused API that has been removed in the new version	Pre build	No	Yes	Running the build process is a simple way to find these removed APIs
6	Stop Using Deprecated APIs	Several APIs are declared as deprecated, and may be removed in the next release. This step is optional, if it is required to stop using these APIs, and start using the alternatives	Development	No	No	Performing this step will save the need to handle the “deprecated APIs removal” step in the next version
7	Modify Changed APIs	Several APIs were removed from this version and cannot be used anymore. This step is mandatory and an alternative implementation is available	Development	No	Yes	Running the build process is a simple way to find these removed APIs
8	New APIs	Several new APIs are introduced and should be used for new features that replace temporary patches given in earlier versions.	Development	No	Yes	
9	Update Configuration Changes	In case that the PE is loaded through customization, when using the configuration file, new attributes need to be added to the configuration file.	Development	No	Yes, if the files were previously customized	

#	Activity	Description	Scheduling	Automatic Process	Prerequisite for Development	Comments
10	Extract Data	Some of the data on which PE is depended should be re-extracted.	Development	No	Yes	Customer offers & Customer parameters should be fully extracted due to the addition of new core attributes

Regenerate Valid Values

Delete all old core domains in the customization area. Add all domains VVs via Fox.

All customized Fox objects need to be regenerated prior to performing the upgrade.

Regenerate Domains

Regenerate all customized domains.

All customized domain objects need to be regenerated prior to performing the upgrade.

Modify Changed Database Structures

Database changes are described in the Pricing Engine Release Notes. The automatic database upgrade process is responsible for the structure and data upgrade. The data upgrade is required when there is a need to upgrade a live environment containing data from 5.5 to 6.0.

The Tiger tool supports running the upgrade.

As database objects may be used by customizations (e.g., map customization views to core table fields), and removed or modified objects may impact the customization layer, **these changes are listed here in addition to the Release Notes.**

DB Object Area & GDD BB	Object Name	Object Type	Object Change Type	Column Name	Column Change Type	Domain	Change Description
pmgdd600 V64	PM1_CYCLE_CHANGE	Table	New	CUSTOMER_ID	New	NUMBER(9)	New table
pmgdd600 V64	PM1_CYCLE_CHANGE	Table	New	EFFECTIVE_DATE	New	DATE	New table
pmgdd600 V64	PM1_CYCLE_CHANGE	Table	New	NEW_CYCLE_CODE	New	NUMBER(2)	New table
pmgdd600 V64	PM1_CYCLE_CHANGE	Table	New	OLD_CYCLE_CODE	New	NUMBER(2)	New table
pmgdd600 V64	PM1_CYCLE_CHANGE	Table	New	STATUS	New	CHAR(1)	New table
pmgdd600 V64	PM1_CYCLE_CHANGE	Table	New	SYS_CREATION_DATE	New	DATE	New table
pmgdd600 V64	PM1_CYCLE_CHANGE	Table	New	SYS_UPDATE_DATE	New	DATE	New table
pmgdd600 V64	PM1_CYCLE_CHANGE	Table	New	APPLICATION_ID	New	CHAR(6)	New table
pmgdd600 V64	REJECTED_EVENT	Table	Modify	ERROR_CODE	Field size increased	VARCHAR2(8)	Field size increased

PM1_CYCLE_CHANGE Table

This is a new table defined in the Pricing Engine as part of the Change Cycle functionality. When using previous solutions for the Change Cycle functionality, given as part of defect #12697 patch #9081, it is required to convert the previously stored data to the PM1_CYCLE_CHANGE table.

Remove APIs

The following deprecated objects/methods were removed from this version. In order to adjust the customized code to this change, it is required to stop using these objects and methods.

Class Name	Method Name	Type	Removal Reason	Alternative
PricingEngine	initRerate	Method removal	The functionality was moved to the regular init method	Use the PricingEngine init method (full initialization) and then call the new setCycleInfo method for each processed cycle
UsageQueryAPI	init	Method removal	Functionality was moved to Billing	Use new APIs from Billing
UsageQueryAPI	release	Method removal	Functionality was moved to Billing	Use new APIs from Billing
UsageQueryAPI	getRatedEvents	Method removal	Functionality was moved to Billing	Use new APIs from Billing
UsageQueryAPI	getRatedEventDetails	Method removal	Functionality was moved to Billing	Use new APIs from Billing
UsageQueryAPI	getPIs	Method removal	Functionality was moved to Billing	Use new APIs from Billing
UsageQueryAPI	getPIDetails	Method removal	Functionality was moved to Billing	Use new APIs from Billing
EntityCursor	getErrorInfo	Method removal	The method was removed since the logic is now done by the IO F/W which doesn't output the TT or Oracle errors	Log file will contain the TT or Oracle errors

A simple build process easily generates a list of all areas in the customized code that need to be fixed.

Stop Using Deprecated APIs

The following APIs were declared as deprecated in this version and should not be used in the long term. As deprecated APIs can be removed from future versions, it is highly recommended to stop using these APIs.

Class Name	Method Name	Type	Alternative
Pricing Engine	initDataDictionary	Deprecated	Use the new initDataDictionary method that uses the InitParameters input structure

Modify Changed APIs

The following objects/methods were removed from or changed in this version. In order to adjust the customized code to this change, it is required to stop using or change the use of these objects and methods.

Class Name	Method Name	Type	Removal/Change Reason	Alternative
Pricing Engine	prepareConnection	Method changed	DBConnection input parameter was changed to generic IO F/W connection object acm::common::io:IConnection	Change the use of the prepareConnection method to receive as input a pointer to one of the following changed classes: OracleConnection, TimesTenConnection or DualConnection
Pricing Engine	initImplementationRepository	Method changed	Method was changed to receive an optional input parameter, UtString& i_encoding.	The additional encoding parameter is optional and will be set to default iso8859-1 if not supplied to the method. It is required to pass the correct encoding in any case where the system works with XMLs encoded with anything other then iso8859-1.

A simple build process easily generates a list of all areas in the customized code that need to be fixed.

New APIs

The following objects/methods were added to this version. In order to adjust the customized code to this change, it is required to start using these objects and methods.

Class Name	Method Name	Type	Replacement Reason	Alternative
Pricing Engine	createCycleChange	New	This new method is used in order to create cycle change entities. It is part of the cycle change feature and should be used instead of the partial solution given for defect #12697 patch #9081	
Pricing Engine	setCycleInfo	New	Used for setting the cycle info for re-rate processing. This method is called after calling the init method, for setting the PE to work in re-rate mode	
Pricing Engine	setReferenceDataStorage	New	This method is used for setting the Pricing Engine reference data storage object for the current processing thread (for Multi-threaded processing only). If the input storage object is null, then the Pricing Engine automatically sets the storage to the latest existing storage available	
Pricing Engine	releaseReferenceDataStorage	New	This method releases the current processing thread storage object (for Multi-threaded processing only).	

Update Configuration Changes

This section details all configuration changes that were made in the core layer that impact the customization layer. Configuration properties are maintained in properties files and other various configuration files.

pmPricingEngineConfig.xml Changes

The Pricing Engine configuration file was added with the following new tags:

- ParsingParameters – XML parsing related attributes.
- EventTableStatus – indicates the state of the event table. Can be one of the following values: HYBRID, FIELDED and DYNAMIC.
  If the parameter is not passed as part of the configuration the PE sets the parameter to the default value: DYNAMIC.
- RunMode – indicates the PE run mode in relation to the "PIT with no PI" functionality. The possible values for the parameter are Always, OfflineRating and OnlineRating.
  If the parameter is not passed as part of the configuration, PE will activate all the handlers regardless of the run mode.

The Pricing Engine configuration file was added with the following new attributes:

- PartitionID – indicates the partition on which the PE instance is running. Each PE instance can be connected/run on one partition at a time. The PartitionID should be provided to the PE as part of the configuration file.
  If the parameter is not passed as part of the configuration, PE assumes that only one partition exists.
  The location in the configuration file of the new PartitionID attribute is under the EventProcessingParameters tag.
- MaxLoopIterations – indicates the maximum number of loops, on complex attributes, that the PE instance can perform.
  If the parameter is not passed as part of the configuration, PE sets the maximum to a default of 20 iterations.
  The location in the configuration file of the new MaxLoopIterations attribute is under the EventProcessingParameters tag.
- encoding – indicates the encoding the PE will use for parsing the XML files (Implementation Repository).
  If the parameter is not passed as part of the configuration, PE sets the attribute to a default value of iso8859-1.
  The location in the configuration file of the new encoding attribute is under a new tag: ParsingParameters.
- MaxPrefetchRows – indicates the maximum number of rows that will be fetched at once from the DB (one round trip). This attribute sets the maximum number of rows that will be fetched form the PM1_CUSTOMER_OFFER_PARAMS table.
  If the parameter is not passed as part of the configuration, PE sets the attribute to a default value of 20.
  The maximum configurable limitation is 128. If the parameter is set to any number greater then 128, the PE will automatically set it to the maximum of 128.

Extract Data

The following attributes were added as core attributes requiring a full Customer Offers and Customer Parameters extract before the Pricing Engine can be run. The attributes themselves will be added to the implementation repository as part of the PC conversion.

Entity	Attribute	Type	Addition Reason	Alternative
Customer Parameters	Change cycle indicator	Core	Attribute was added for supporting the new Cycle Change functionality introduced in this version	
Customer Offers	Effective activity code	Core	Attribute was added for supporting the new Move Subscriber functionality introduced in this version	
Customer Offers	Expiration activity code	Core	Attribute was added for supporting the new Move Subscriber functionality introduced in this version	

Full extract of Customer Offers and Customer Parameters should be performed.

Extension Functions

Pricing Engine is now using IO Framework for all DB related operations. All core extension functions requiring access to the DB were modified to use IO Framework. It is suggested that all customization extension functions, that are required to perform DB operations, will also use IO Framework.

Replenishment Management

Summary of Upgrade Activities

The following table summarizes the activities that are involved in the Development upgrade process. The activities are presented in the recommended order of their execution. Further details about each activity are described in subsequent sections.

Activity	Description	Scheduling	Automatic Process	Prerequisite for Development	Comments
Regenerate Valid Values	Delete all old core domains in the customization area, add all domains DVMs via FOX	Development	No	No	The old solution is applicable, but not recommended
Adjust Application Make Files	Adjust the customized make files to the new version	Pre Build	No	No	
Regenerate Domains	Regenerate all customized domains	Build	Yes	Yes	
Regenerate Data Types	Rebuild the customization area	Build	Yes	Yes	
Change DVM files format	Run DVM migration script	Development	No	Yes	The DVM is now part of the dStudio, the format change is mandatory to comply with the studio standard. See Appendix B: DVM, RDM Persistence Tools - Migration Script
Regenerate Database Views (JUTIL 5.2 Packages)	Regenerate all customized views	Pre Build	Semi (need to check in files manually)	Yes	
Migrate Reference datalayers to DVM	Run DVM migration script	Development	No	Yes	. See Appendix B: DVM, RDM Persistence Tools - Migration Script
Regenerate Reference Data Layers (JUTIL 5.2 Packages)	Regenerate all customized data layers	Pre Build	Semi (need to check in files manually)	Yes	
Update Message Handling Services (JUTIL 5.2 Packages)	Check if code changes are needed due to interface change	Pre Build	No	Yes	

Activity	Description	Scheduling	Automatic Process	Prerequisite for Development	Comments
Modify Changed Database Structures					
Remove APIs	Adjust any code that uses a deprecated API that has been removed in the new version	Pre build	No	Yes	Running the build process is a simple way to find these removed APIs
Stop Using Deprecated APIs	Several APIs are declared as deprecated, and may be removed in the next release. This step is optional, if it is required to stop using these APIs, and start using the alternatives	Development	No	No	Performing this step will save the need to handle the "deprecated APIs removal" step in the next version

Regenerate Valid Values

All customized FOX objects need to be regenerated prior to performing the upgrade.

Adjust Application Make Files

The following table summarizes the customized Make file changes that are relevant for version 6.0.

Item	Change Type	Description	Required Steps
SONAR template builder	New	Generator which builds SONAR template files from input application properties files	Add it to your Make file. Make sure that your properties comply with the SONAR tags standard
Building Block GRPL	Deleted	The building block was deleted. All application scripts were moved to GRPLE.	N/A
Building Block GRPGUI	Deleted	The building block was deleted. The new RPL testing GUI is the Method Invoker.	N/A

Regenerate Domains

All customized domain objects need to be regenerated prior to performing the upgrade.

JUTIL 5.2 Packages

Regenerate Database Views

All customized DVM views need to be upgraded and regenerated prior to performing the upgrade. See Appendix B: DVM, RDM Persistence Tools - Migration Script

Regenerate Reference Data Layers

All RVM customization views need to be regenerated prior to performing the upgrade.

Modify Changed Database Structures

Database changes are described in the Replenishment Management Release Notes. The automatic database upgrade process is responsible for the structure and data upgrade. The data upgrade is required when there is a need to upgrade a live environment containing data from 5.5 to 6.0.

The Tiger tool supports running the upgrade.

As database objects may be used by customizations (e.g., map customization views to core table fields), and removed or modified objects may impact the customization layer, **these changes are listed here in addition to the Release Notes.**

DB Object Area & GDD BB	Object Name	Object Type	Object Change Type	Column Name	Column Change Type	Domain	Change Description
grplgdd	RPM_SUBSCRIBER	Table	Modify	PRIM_RESOURCE_VAL	Field Size Increased	VARCHAR2(63)	Field Size Increased
grplgdd	RPM_SUBSCRIBER	Table	Modify	IMSI	Field Size Increased	VARCHAR2(63)	Field Size Increased
grplgdd	RPM_RECHARGES	Table	Modify	PRIM_RESOURCE_VAL	Field Size Increased	VARCHAR2(63)	Field Size Increased
grplgdd	RPM_RECHARGES	Table	Modify	IMSI	Field Size Increased	VARCHAR2(63)	Field Size Increased
grplgdd	RPM_MEMO	Table	Modify	MEMO_SYSTEM_TXT	Field Size Increased	VARCHAR2(1000)	Field Size Increased
grplgdd	RPM_MEMO	Table	Modify	MEMO_ADDITIONAL_TXT	Field Size Increased	VARCHAR2(1000)	Field Size Increased
grplgdd	RPM_RCG_ERROR	Table	Modify	PRIM_RESOURCE_VAL	Field Size Increased	VARCHAR2(63)	Field Size Increased
grplgdd	RPM_RCG_ERROR	Table	Modify	IMSI	Field Size Increased	VARCHAR2(63)	Field Size Increased

DB Object Area & GDD BB	Object Name	Object Type	Object Change Type	Column Name	Column Change Type	Domain	Change Description
grplgdd	RPM_BUCKET_ID	Table	Modify	IMSI	Field Size Increased	VARCHAR2(63)	Field Size Increased
grplgdd	RPM_CONF_RCG_MTD	Ref Table	Modify	DATATYPE	Field Size Increased	VARCHAR2(60)	Field Size Increased
grplgdd	RPM_CONF_CHANNEL	Ref Table	Modify	DATATYPE	Field Size Increased	VARCHAR2(60)	Field Size Increased
grplgdd	RPM_MEMO_TYPE	Ref Table	Modify	MEMETP_SYSTEM_TXT	Field Size Increased	VARCHAR2(1000)	Field Size Increased
grplgdd	RPM_MARKET	Ref Table	Deleted				Deleted
grplgdd	RPM_CHANNEL	Ref Table	Deleted				Deleted
grplgdd	RPM_BILL_CHARGE_DET	Ref Table	Deleted				Deleted
grplgdd	RPM_MARKET_POLICY	Ref Table	Deleted				Deleted
grplgdd	RPM_PLC_PROFILE	Ref Table	Deleted				Deleted
grplgdd	RPM_METHODS	Ref Table	Deleted				Deleted
grplgdd	RPM_BAL_POLICY	Ref Table	Deleted				Deleted
grplgdd	RPM_CONF_RCG_MTD.IDAT	IDAT	Deleted				Deleted
grplgdd	RPM_PLC_PROFILE.IDAT	IDAT	Deleted				Deleted
grplgdd	RPM_BILL_CHARGE_DET.IDAT	IDAT	Deleted				Deleted
grplgdd	RPM_VLD_CHN_TO_RCG.IDAT	IDAT	Deleted				Deleted
grplgdd	RPM_CONF_CHANNEL.IDAT	IDAT	Deleted				Deleted

Remove APIs

N/A

Stop Using Deprecated APIs

The following APIs were declared as deprecated in this version and should not be used in the long term. As deprecated APIs can be removed from future versions, it is highly recommended to stop using these APIs. The Deprecated view in the Javadoc HTML documentation lists all the deprecated methods.

Class Name	Method Name	Alternative
RPL1Api	activateSubscriber	Class: RPL1SubscriberServices Method: activateSubscriber
RPL1Api	addAdditionalBonus	Class: RPL1BucketServices Method: addAdditionalBonus
RPL1Api	addInstalment	Class: RPL1BucketServices Method: addInstalment
RPL1Api	adjustLeftAmount	Class: RPL1BucketServices Method: adjustLeftAmount
RPL1Api	cancelChangeBillCycle	Class: RPL1BucketServices Method: cancelChangeBillCycle
RPL1Api	cancelRecharge	Class: RPL1RechargeServices Method: cancelRecharge
RPL1Api	cancelSplitRecharge	Class: RPL1RechargeServices Method: cancelSplitRecharge
RPL1Api	cancelSubscriber	Class: RPL1SubscriberServices Method: cancelSubscriber
RPL1Api	changeBillCycle	Class: RPL1BucketServices Method: changeBillCycle
RPL1Api	changeBucket	Class: RPL1BucketServices Method: changeBucket
RPL1Api	changeCycleFirstRun	Class: RPL1BatchServices Method: changeCycleFirstRun
RPL1Api	changeCycleSecondRun	Class: RPL1BatchServices Method: changeCycleSecondRun
RPL1Api	changePcnName	Class: RPL1BucketServices Method: changePcnName
RPL1Api	changePolicy	Class: RPL1BucketServices Method: changePolicy
RPL1Api	changePrimaryResource	Class: RPL1SubscriberServices Method: changePrimaryResourceInfo
RPL1Api	changeSubscriberPaymentCategory	Class: RPL1SubscriberServices Method: changeSubscriberPaymentCategory
RPL1Api	closeBucket	Class: RPL1BucketServices Method: closeBucket
RPL1Api	connectSubscriber	Class: RPL1SubBucketServices Method: changeEventDistribution
RPL1Api	disconnectSubscriber	Class: RPL1SubBucketServices Method: changeEventDistribution
RPL1Api	extendExpirationDate	Class: RPL1RechargeServices Method: changeExpDate

Class Name	Method Name	Alternative
RPL1Api	getAllPPSBalanceDetails	Class: RPL1BucketServices Method: getOLCBalanceInfo
RPL1Api	getBalanceByPcn	Class: RPL1BucketServices Method: getOLCBalanceInfo
RPL1Api	getBalanceByPCN	Class: RPL1BucketServices Method: getOLCBalanceInfo
RPL1Api	getBalanceExpirationDate	Class: RPL1BucketServices Method: getBucketHeader
RPL1Api	getPCNDetails	Class: RPL1BucketServices Method: getBucketHeader
RPL1Api	getPPSBalanceAmount	Class: RPL1BucketServices Method: getOLCBalanceInfo
RPL1Api	getProfileParamVal	Class: RPL1RefDataMngr Method: getProfileParamVal
RPL1Api	getRegularBonusPolicy	Class: RPL1BucketServices Method: getRegularBonusPolicy
RPL1Api	getVoucherAmount	Class: RPL1RechargeServices Method: getVoucherAmount
RPL1Api	handleTriggerRequest	Class: RPL1RechargeServices Method: handleTriggerRequest
RPL1Api	newBucket	Class: RPL1BucketServices Method: newBucket
RPL1Api	openBucket	Class: RPL1BucketServices Method: openBucket
RPL1Api	preActivation	Class: RPL1SubscriberServices Method: preActivation
RPL1Api	rechargeService	Class: RPL1RechargeServices Method: rechargeService
RPL1Api	removeAdditionalBonus	Class: RPL1BucketServices Method: removeAdditionalBonus
RPL1Api	removeInstalment	Class: RPL1BucketServices Method: removeInstalment
RPL1Api	restoreSubscriber	Class: RPL1SubscriberServices Method: restoreSubscriber
RPL1Api	resumeSubscriber	Class: RPL1SubscriberServices Method: resumeSubscriber
RPL1Api	rpmRechargeService	Use APInvoker as a Subscriber daemon for RPL instead of TRB daemon
RPL1Api	rpmRegisterService	Use APInvoker as a Subscriber daemon for RPL instead of TRB daemon
RPL1Api	rpmService	Use APInvoker as a Subscriber daemon for RPL instead of TRB daemon
RPL1Api	saleEvent	Class: RPL1SubscriberServices Method: saleEvent
RPL1Api	setExpirationDate	Class: RPL1RechargeServices Method: changeExpDate
RPL1Api	splitRechargeByPCN	Class: RPL1RechargeServices Method: splitRecharge

Class Name	Method Name	Alternative
RPL1Api	splitRechargeBySubscriber	Class: RPL1RechargeServices Method: splitRecharge
RPL1Api	suspendSubscriber	Class: RPL1SubscriberServices Method: suspendSubscriber
RPL1Api	triggerSplitRechargeByPCN	Class: RPL1RechargeServices Method: triggerSplitRecharge
RPL1Api	triggerSplitRechargeBySubscriber	Class: RPL1RechargeServices Method: triggerSplitRecharge
RPL1Api	updateCustomerId	Class: RPL1BucketServices Method: updateCustomerId
RPL1Api	updateSubscriberDetails	Class: RPL1SubscriberServices Method: changeSubscriberGeneralInfo
RPL1Api	viewAllAddBonusDetails	Class: RPL1RefDataMngr Method: viewAllAddBonusDetails
RPL1Api	viewAllInstallmentDetails	Class: RPL1RefDataMngr Method: viewAllInstallmentDetails
RPL1Api	viewAllPoliciesDetails	Class: RPL1RefDataMngr Method: viewAllPoliciesDetails
RPL1Api	viewAllSubscriberBalanceDetails	Class: RPL1SubscriberServices Method: viewAllSubscriberBalanceDetails
RPL1Api	viewConfChannelDetails	Class: RPL1RefDataMngr Method: viewConfChannelDetails
RPL1Api	viewConfRchargeMethodDetails	Class: RPL1RefDataMngr Method: viewConfRchargeMethodDetails
RPL1Api	viewOpenAccmulation	Class: RPL1BucketServices Method: getBucketAccmulations
RPL1Api	viewOpenAddBonusAccDetails.	Class: RPL1BucketServices Method: getBucketAccmulations
RPL1Api	viewOpenInstallmentAccDetails	Class: RPL1BucketServices Method: getBucketAccmulations
RPL1Api	viewPolicyDetails	Class: RPL1BucketServices Method: viewPolicyDetails
RPL1Api	viewRecharge	Class: RPL1RechargeServices Method: viewRecharge
RPL1Api	viewRecharge	Class: RPL1BucketServices Method: viewRecharge
RPL1Api	viewRechargeError	Class: RPL1BucketServices Method: viewRechargeError
RPL1Api	viewVldChnToRcgDetails	Class: RPL1RefDataMngr Method: viewVldChnToRcgDetails

Update Operational Changes

Operational Jobs

The following table summarizes all the Operational jobs that are new or changed in this version:

Functionality	Job Name	Script	Change Type	Dependency	Description
RPL Daemon for subscribed events	RPL1TRBSUB	rpl1trbsub_Sh	Deleted	None	Replaced by the APInvoker.
RPL Refresh Job	RPL1REFRESH	rpl1refresh_Sh	Deleted	None	Replaced by the 24X7 EJ refresh agent.

Operational Parameter Tables

The RPL1ENDOFDAY job has a new optional parameter for defining the unique ongoing run number.

On the first daily run of this job this parameter should be left as null. The job will then retrieve a new sequence for identifying this daily run.

In case of a daily rerun, because of errors, setting this parameter with the daily run unique ID will cause the job to skip the buckets it has already updated, and, by this, will improve the performance of the job.

Job Dependencies

The following table summarizes changes to the Operational job dependencies in this version:

Map Name: End of day

Job Name	Possible Impact and Comments
RPL1RECOVERY	This daemon should be stopped before the reference refresh operation, and then started again.

See the RPL Run Book for more details regarding this job's parameter.

Integrate with SONAR

The customization configuration files have to be modified to use the SONAR standard. The following user defined variables should be set on installation:

- Number of FR servers (default is 1)
- FR(OLC) host IP Address or name.
- FR(OLC) port number.
- Voucher Manager Connection Indicator.
- Voucher Manager Host name or IP address (mandatory only if the Voucher Manager Connection Indicator was set to "Y")
- Voucher Manager Port Number (mandatory only if the Voucher Manager Connection Indicator was set to "Y")

Adjust Batch Processes

Batch processes now have customization exit points, both standard DVM (database related) and business, where applicable.

Adjust Daemon Processes

The recovery daemon is controlled using the AMC online.

Update Configuration Changes

This section details all configuration changes that were made in the core layer that impact the customization layer. Configuration properties are maintained in properties files and other various configuration files.

Properties Files

Properties files from older versions were replaced with new properties files according to the Java Foundation (JF) standards.

The new properties files are:

- RPL1App.proeprties – Contains all RPL application properties for both EJB and batch modes. Some properties were moved from the RPM_PROFILE table to this file and are set during the installation.
- RPL1AppBatch.proeprties – Contains RPL batch specific properties.
- RPL1JF.proeprties – Contains standard JF properties.
- RPL1JFBatch.proeprties – Contains standard JF properties for batch jobs and daemons.
- RPL1JFEJB.proeprties – Contains standard JF properties for EJB mode.

See the Properties/Implementation Changes section in the RPL Release Notes document. In addition see SONAR changes section in this document.

Other Configuration Files Changes

N/A

Update XML API XSDs

Starting with version 6.0, RPL is integrated with the APInvoker, and uses it as the subscriber application for the RPL events, from CM, Billing and OLC.

All Subscribed XSDs are generated using the XSD API Generator.

Update Customization Interface

Customization Public Objects

Recharge Customization

In the event that it is desired to customize a core flow, or create a new flow, in the Customization layer in such a way that some or most core steps will still be executed, the Recharge flows enable any flow to register a parent flow, and, by this, use its core on the flow's execution.

For example:

A new flow, called *CustVoucherRechargeFlow*, is created in the Customization layer. The new flow is accessing a different Voucher Manager system than the core RPL flow. The new flow will register the core RPL flow as a parent flow, which means that in runtime, the core steps are automatically executed, unless overridden in exit points.

Common Handlers

RPL has some internal utilities that are available for customization:

- Accumulations Modifier

  Enables the customization layer to modify the core behavior regarding accumulations opened in RPL, e.g., Installments, Bonuses.

- Policy Controller

  Enables the customization layer to access the RPL policies utilities.

See the RPL Customization Guide for more details.

Exit Points Changes

All the special Recharge exit points that required the Customization layer to override core XMLs have been **removed**, and replaced by regular exit points, where necessary.

Exit points that were in use in previous versions must be rewritten in the new standard and registered in the new customization handlers table: RPL1_HANDLER_CUSTOMIZE.

For more details see the RPL Customization Guide

Automatic Upgrade Scripts

The following implementation data scripts should run **after** the upgrade.

All scripts are located under:

- Building Block: grplgdd
- Topic: data/src

Script Location and Name	Execution Order
ABP_UPG_rplV550_V600_RPL1_OLC_MESSAGES.sql	1
ABP_UPG_rplV550_V600_RPM_PROFILE.sql	2
ABP_UPG_rplV550_V600_RPM_CONF_CHANNEL.sql	3
ABP_UPG_rplV550_V600_RPM_CONF_RCG_MTD.sql	4
ABP_UPG_rplV550_V600_RPM_ACTIVITY.sql	5
ABP_UPG_rplV550_V600_RPM_VLD_CHN_TO_RCG.sql	6
ABP_UPG_rplV550_V600_rpm_activity_update_new_fields.sql	7
ABP_UPG_rplV550_V600_rpm_vld_chn_to_rcg_update_parent_key.sql	8
ABP_UPG_rplV550_V600_rpl1_cycle_instance.sql	9***

The <RPL application ID in CC> and <Unique string> will be provided by the DBAs at the time of upgrade.

*** This script must be executed after Billing was executed in the upgrade process (post upgrade).

Resource Management

Summary of Upgrade Activities

The following table summarizes the activities that participate in the development upgrade process. The activities are presented in the recommended order of their execution. Further details about each activity are described in subsequent sections.

#	Activity	Description	Scheduling	Automatic Process	Prerequisite for Development	Comments
1	Regenerate Valid Values	Delete all old core domains in the customization area, add all domains VVs via FOX	Development	No	No	The old solution is applicable, but not recommended
2	Adjust Application Make Files	Adjust the customized make files to the new version	Pre Build	No	No	
3	Regenerate Domains	Regenerate all customized domains	Build	Yes	Yes	
4	Regenerate Data Types	Rebuild the customization area	Build	Yes	Yes	
5	Change DVM files format	Run DVM migration script	Development	No	Yes	The DVM is now part of the dStudio, the format change is mandatory to comply with the studio standard. See Appendix B: DVM, RDM Persistence Tools - Migration Script
6	Regenerate Database Views	Regenerate all customized views	Pre Build	Semi (need to check in files manually)	Yes	
7	Migrate Reference datalayers to DVM	Run DVM migration script	Development	No	Yes	. See Appendix B: DVM, RDM Persistence Tools - Migration Script
8	Regenerate Reference Data Layers (JUTIL 5.2 Packages)	Regenerate all customized data layers	Pre Build	Semi (need to check in files manually)	Yes	
9	Update Message Handling Services)	Check if code changes are needed due to interface change	Pre Build	No	Yes	
10	Modify Changed Database Structures					

#	Activity	Description	Scheduling	Automatic Process	Prerequisite for Development	Comments
11	Manual reference Data Configuration	Adjust any code that uses a deprecated API that has been removed in the new version	Pre build	No	Yes	Running the build process is a simple way to find these removed APIs
12	Stop Using Deprecated APIs	Several APIs are declared as deprecated, and may be removed in the next release. This step is optional, if it is required to stop using these APIs, and start using the alternatives	Development	No	No	Performing this step will save the need to handle the "deprecated APIs removal" step in the next version
13	Modify Changed APIs					
14	Integrate with SONAR	Modify customization configuration files to use the SONAR standard	Development	No	No	
15	Adjust Batch Processes	Integration with SONAR. Adjust the modified core's scripts Operational setup	Development	No	Yes	
16	Update Customization Interface	In case that the PC CM Load configuration files were customized in the previous version, merge the changes with the new files, delivered with this version	Development	No	Yes, if the files were previously customized	
17	All TopLink configuration files were removed	Generate new APIs	Development	Yes	No	
18	Automatic Upgrade Scripts	Script to run in order to perform the upgrade in automatic manner	Defined by the script's context	Yes	Defined by the script's context	Scripts that replace the manual upgrade activities, e.g. data migration, for XML's, DB, operational, shell scripts etc.

Regenerate Database Views

All customized DVM views need to be upgraded and regenerated prior to performing the upgrade. See Appendix B: DVM, RDM Persistence Tools - Migration Script

Regenerate Reference Data Layers (JUTIL 5.2 Packages)

All RDM customization datalayers need to be migrated and regenerated prior to performing the upgrade. See Appendix B: DVM, RDM Persistence Tools - Migration Script

Update Message Handling Services

The RM messages class: amdocs.rm3g.messages.URM1Messages has been replaced with the following: adocs.rm3g.exceptions.ErrorMessages.

Modify Changed Database Structures

Database changes are described in the Resource Management Release Notes. The automatic database upgrade process is responsible for the structure and data upgrade. The data upgrade is required when there is a need to upgrade a live environment containing data from version 5.5 to version 6.0.

The Tiger tool supports running the upgrade.

As database objects may be used by customizations (e.g., map customization views to core table fields), and removed or modified objects may impact the customization layer, **these changes are listed here in addition to the Release Notes.**

DB Object Area & GDD BB	Object Name	Object Type	Object Change Type	Column Name	Column Change Type	Domain	Change Description
URMGDD - APP	RM1_BULK	Table	New GTT Table				New GTT table, which helps in loading the Unified Resources from the Data Base to memory in Bulk operations
URMGDD - APP	RM1_REQUEST	Table	New				A new table, which stores all the Unified Resources bulk requests and the end state Requests.
URMGDD - APP	RM1_SEQ_ARRAY	Table	New				A new table, which helps in the Pre Sequence Allocation mechanism. This table is populated by the core (IDAT).
URMGDD - APP	RM1_PACKAGE	Table	Rename				Replace: RM_PACKAGES

DB Object Area & GDD BB	Object Name	Object Type	Object Change Type	Column Name	Column Change Type	Domain	Change Description
URMGDD - APP	RM1_PACKAGE_HISTORY	Table	Rename				Replace: RM_PACKAGES_HIST
URMGDD - APP	RM1_PACKAGE_ATTRIBUTES	Table	Rename				Replace: RM_PACKAGE_ATTR
URMGDD - APP	RM1_PACKAGE_CONTENT	Table	Rename				Replace: RM_PACKAGE_CONTENT
URMGDD - APP	RM1_RESOURCE	Table	Rename				Replace: RM_RESOURCE_STOCK
URMGDD - APP	RM1_RESOURCE_HISTORY	Table	Rename				Replace: RM_RESOURCE_STK_HIST
URMGDD - APP	RM1_RESOURCE_ATTRIBUTES	Table	Rename				Replace: RM_RESOURCE_ATTR
URMGDD - APP	RM_RESOURCE_CATEGORY	Table	Delete				Replaced by a new column CATEGORY_INDEX on the RM1_RESOURCE, and the new reference table: RM1_CATEGORY_ASSIGNMENTS
URMGDD - APP	RM_SEQUENCES	Table	Delete				Replaced by the new Sequence Allocation table RM1_SEQ_ARRAY
URMGDD - APP	RM_PACKAGE_CNT_HIST	Table	Delete				No need for this table
URMGDD - REF	RM1_CATEGORY_ASSIGNMENTS	Table	New				A new table which defines all the resource category combinations per resource type
URMGDD - REF	RM1_END_STATUS	Table	New				A new table which is used by the end state functionality

DB Object Area & GDD BB	Object Name	Object Type	Object Change Type	Column Name	Column Change Type	Domain	Change Description
URMGDD - REF	RM_LIFE_CYCLE_ACT	Table	Modify	RLCA_ASSIGN_CAT_IND	New	YESNOIND	Indicates whether the activity should call the resource category assignment process
URMGDD - REF	RM_LIFE_CYCLE_ACT	Table	Modify	RLCA_POOL_TREATMENT	New	RM1POOLTRT	The required pool treatment for this activity. The valid values are: SET - Set the resource pool according to the Target Pool activity parameter; CLEAR - Clear the resource pool assignment; NA - The pool treatment is not relevant for the activity.
URMGDD - REF	RM_PACKAGE_TP_ATTR	Table	Modify	RPTA_SRCH_KEY1_IND	New	YESNOIND	Defines weather the current attribute value should be maintained in the Packages' SearchKey1 field. (In order to allow quick search according to this attributes value)
URMGDD - REF	RM_PACKAGE_TYPE	Table	Modify	RPT_PATITION_KEY	New	ENTSEQNO	Defines the Unified Resource Type partition Key (This value is crucial, since the first partition level is set according to this value)
URMGDD – REF	RM_RESOURCE_TYPE	Table	Modify	RRT_PATITION_KEY	New	ENTSEQNO	Defines the Unified Resource Type partition Key (This value is crucial, since the first partition level is set according to this value)
URMGDD – REF	RM_PACKAGE_TYPE_CONT	Table	Modify	RPTC_GEN_BY_CMP_TP	New	RMNAME	Indicates weather the package value is populated according to the current Component's value. Y - The package value is populated according to the current component's value. N or NULL - The package value is not affected by this component. (This value is crucial, since the value of the package is determined according to it. It must be populated for all the packages components)
URMGDD – REF	RM_PACKAGE_TYPE_CONT	Table	Modify	RPTC_POPULATE_IND	New	YESNOIND	Contains the entity type id according to which the value will be generated, in case that the current component should be generated according to another component's value in the package (in case that more than one instance is allowed from the current component type, an arbitrary value is picked)

DB Object Area & GDD BB	Object Name	Object Type	Object Change Type	Column Name	Column Change Type	Domain	Change Description
URMGDD – REF	RM_DICTIONALRY	Table	Delete IDAT records	(RD_ID,RD_NAME,RD_VALUE) = (12, 'ACTIVITY','UNLOCK')	Delete		The Resource Management removed some hard coded activities and statuses.
URMGDD – REF	RM_DICTIONALRY	Table	Delete IDAT records	(RD_ID,RD_NAME,RD_VALUE) = (11, 'ACTIVITY','LOCK')	Delete		The Resource Management removed some hard coded activities and statuses.
URMGDD – REF	RM_DICTIONALRY	Table	Delete IDAT records	(RD_ID,RD_NAME,RD_VALUE) = (10,'STATUS','LOCKED')	Delete		The Resource Management removed some hard coded activities and statuses.
URMGDD – REF	RM_DICTIONALRY	Table	Delete IDAT records	(RD_ID,RD_NAME, RD_VALUE) = (9, 'STATUS', 'PACKED')	Delete		The Resource Management removed some hard coded activities and statuses.
URMGDD – REF	RM_DICTIONALRY	Table	Delete IDAT records	(RD_ID,RD_NAME, RD_VALUE) = (8, 'STATUS', 'UN_PACKED')	Delete		The Resource Management removed some hard coded activities and statuses.
URMGDD – REF	RM_DICTIONALRY	Table	Delete IDAT records	(RD_ID,RD_NAME, RD_VALUE) = (7, 'STATUS', 'ASSIGNED')	Delete		The Resource Management removed some hard coded activities and statuses.
URMGDD – REF	RM_DICTIONALRY	Table	Delete IDAT records	(RD_ID,RD_NAME, RD_VALUE) = (4, 'ACTIVITY', 'CREATE')	Delete		The Resource Management removed some hard coded activities and statuses.
URMGDD – REF	RM_DICTIONALRY	Table	Delete IDAT records	(RD_ID,RD_NAME, RD_VALUE) = (3, 'ACTIVITY', 'DEASSIGN')	Delete		The Resource Management removed some hard coded activities and statuses.
URMGDD – REF	RM_DICTIONALRY	Table	Delete IDAT records	(RD_ID,RD_NAME, RD_VALUE) = (2, 'ACTIVITY', 'REASSIGN')	Delete		The Resource Management removed some hard coded activities and statuses.
URMGDD – REF	RM_DICTIONALRY	Table	Delete IDAT records	(RD_ID,RD_NAME, RD_VALUE) = (1, 'ACTIVITY', 'ASSIGN')	Delete		The Resource Management removed some hard coded activities and statuses.

Manual Reference Data Configuration

The following reference data configuration should be performed manually:

Package Value Population Indicator

Each package must indicate which one of its components defines the package value. This is defined in the reference table: RM_PACKAGE_TYPE_CONT, in the field: RPTC_POPULATE_IND.

This field is mandatory. Otherwise the creation of packages will not function correctly.

Resource/Package Implementation Classes

If as a result of the upgrade to version 6.0 of Resource Management some of the implementation classes were changed; the new implementation class must be updated per Resource/Package type via the RMC.

Resource/Package Core Implementation

Any project which used the Resource Management version 5.5 core implementation must update the name of the implementation class manually, via RMC, according to the following mapping:

Resource Management Version 5.5 Implementation Class Name	Resource Management Version 6.0 Implementation Class Name
amdocs.rm3g.gn. IPv4Impl	amdocs.rm3g.impclass.resource.implementation. IPv4CustomizationDefaultHandler
amdocs.rm3g.gn.MSISDNImpl	amdocs.rm3g.impclass.resource.implementation. MSISDNCustomizationDefaultHandler
amdocs.rm3g.gn.USERNAMEImpl	amdocs.rm3g.impclass.resource.implementation. USERNAMECustomizationDefaultHandler
amdocs.rm3g.gn.UICCImpl	amdocs.rm3g.impclass.resource.implementation. UICCCustomizationDefaultHandler
amdocs.rm3g.gn.VPIPNv4Impl	amdocs.rm3g.impclass.resource.implementation. VPIPNv4CustomizationDefaultHandler
amdocs.rm3g.gn.IMSIImpl	amdocs.rm3g.impclass.resource.implementation. IMSICustomizationDefaultHandler
amdocs.rm3g.gn.ICCImpl	amdocs.rm3g.impclass.resource.implementation. UICCCustomizationDefaultHandler
amdocs.rm3g.gn.PRVTNUMBERImpl	amdocs.rm3g.impclass.resource.implementation. PRVTNUMBERCustomizationDefaultHandler
amdocs.rm3g.gn.VPNImpl	amdocs.rm3g.impclass.package.implementation. VPNCustomizationDefaultHandler
amdocs.rm3g.gn.USIMImpl	amdocs.rm3g.impclass.resource.implementation. USIMCustomizationDefaultHandler
amdocs.rm3g.gn.SIMImpl	amdocs.rm3g.impclass.resource.implementation. SIMCustomizationDefaultHandler
amdocs.rm3g.gn.KITImpl	amdocs.rm3g.impclass.resource.implementation. KITCustomizationDefaultHandler

Rule Core Implementation

Any project which used the Resource Management version 5.5 core implementation must upgrade to the following:

Resource Management Version 5.5 Implementation Class Name	Resource Management version 6.0 Implementation Class Name
amdocs.rm3g.gn.NumberRangeRuleImpl	amdocs.rm3g.impclass.rules.implementation. NumberRangeRulesImpl
amdocs.rm3g.gn.PatternRuleImpl	amdocs.rm3g.impclass.rules.implementation. NumberPatternRulesImpl

All the core implementation classes are now part of the core project (e.g. gurmsrv_classes.jar).

JNDI Names

All the Resource Management Beans JNDI names were changed according to the following convention: amdocsBeans.RM1<Home Interface Name> E.g. the JNDI name of the Application Manager Bean is: amdocsBeans.RM1ApplManagerConvHome

Table 3: JNDI Names

	JNDI Name	Bean Description
1	amdocsBeans.RM1ApplManagerConvHome	Application Manager (Deprecated)
2	amdocsBeans.RM1LogicalDateHome	Logical Date (Deprecated)
3	amdocsBeans.RM1PackageManagerHome	Package Manager (Deprecated)
4	amdocsBeans.RM1RefreshAgentHome	Refresh Agent (New)
5	amdocsBeans.RM1RequestManagerHome	Request Manager (New)
6	amdocsBeans.RM1UnifiedResourceManagerHome	Unified Resource Manager (New)
7	amdocsBeans.RM1UnifiedResourceServicesHome	Unified Resource Services (New)
8	amdocsBeans.RM1GeneralUtilsHome	General Utilities (New)
9	amdocsBeans.RM1RefDataManagerHome	Reference Data Manager
10	amdocsBeans.RM1ResourceManagerHome	Resource Manager (Deprecated)

Using RM remote ejb object isn't backwards compatible. When using RM remote object the JNDI name should be changed to include RM1.

Remove APIs

The following deprecated objects/methods were removed from this version.

In order to adjust the customized code to this change, it is required to stop using these objects and methods.

Class Name	Method Name	Type	Removal Reason	Alternative
RefDataMngr	public RTCsmRetentionInfo getRTCsmRetentionInfo(String retentionCode, Date activityDate) throws RemoteException	Method removal	The functionality has moved to the NURater	The commitment information is published in the PC export XML files. It can be retrieved from there.

A simple build process easily generates a list of all areas in the customized code that need to be fixed.

Stop Using Deprecated APIs

The following APIs were declared as deprecated in this version and should not be used in the long term. As deprecated APIs can be removed from future versions, it is highly recommended to stop using these APIs. The Deprecated view in Javadoc HTML documentation lists all the deprecated methods.

Class Name	Method Name	Alternative Class Name	Alternative Method Name
ApplManagerConv	All methods		
LogicalDate	public java.sql.Date getLogicalDate(AdminData adminData, LogicalDateType dateChar) throws DataNotFoundException, UrmSystemException, UrmApplicationException, RemoteException;	RMGeneralUtils	public DateInfo getCurrentDateTime(RMSessionData rmSessionData) throws RMException,java.rmi.RemoteException
ResourceManager	public void invokeResourceActivity(AdminData iAdminData, InvokeResourceActData invRscAct) throws RemoteException,DataNotFoundException, UrmSystemException,UrmApplicationException;	RM1UnifiedResourceManager	■ public void invokeUnifiedResourceActivity(RMSessionData RMSessionData,UnifiedResourceActivityInfo unifiedResourceActivity) throws RMException, java.rmi.RemoteException ■ public RMInvokeActivityListStatusInfo invokeUnifiedResourceActivityList(RMSessionData rmSessionDataExt, UnifiedResourceBulkActivityListInfo list) throws RMException, RemoteException;

Class Name	Method Name	Alternative Class Name	Alternative Method Name
ResourceManager	public QuantityProcessedData invokeBulkResourceActivity(AdminData adminData,InvokeResourceBulkActData rscBulkAct) throws RemoteException, DataNotFoundException, UrmSystemException,UrmApplicationException;	RM1UnifiedResourceManager	■ public RMRequestIdInfo invokeBulkActivity(RMSessionData RMSessionData, UnifiedResourceBulkActivityInfo bulkActivity) throws RMException, RemoteException ■ public RMRequestIdInfo invokeBulkActivity(RMSessionData RMSessionData, UnifiedResourceBulkActivityListInfo bulkActivity) throws RMException, RemoteException
ResourceManager	public CategoryValueData [] assignResourceCategories(AdminData adminData, ResourceValueData rscValue) throws RemoteException,DataNotFoundException, UrmSystemException,UrmApplicationException;	NA.	NA
ResourceManager	public ResourceInStockListData getResourceForActivity(AdminData adminData, InvokeResourceBulkActData rscBulkAct) throws RemoteException,DataNotFoundException, UrmSystemException,UrmApplicationException;	RM1UnifiedResourceServices	public UnifiedResourceListInfo getUnifiedResourceForActivity(RMSessionData rmSessionData, UnifiedResourceBulkActivityInfo unifiedResourceBulkActivityInfo, PaginationInfo paginationInfo) throws RMException, RemoteException;

Class Name	Method Name	Alternative Class Name	Alternative Method Name
ResourceManager	public void updateResourceAttributes(AdminData adminData, UpdateResourceAttrData resourceAttrData) throws DataNotFoundException, UrmSystemException, UrmApplicationException, RemoteException;		■ public void UpdateAttributes(RMSessionData rmSessionData,UnifiedResourceAttributesInfo unifiedResourceAttributesInfo) throws RMException,java.rmi.RemoteException ■ public RMInvokeActivityListStatusInfo updateAttributesList(RMSessionData rmSessionData,UnifiedResourceBulkAttributeListInfo list) throws RMException,java.rmi.RemoteException ■ public RMRequestIdInfo updateAttributeBulk(RMSessionData rmSessionData,UnifiedResourceBulkSelectionCriteriaInfo selectionCriteriaInfo,UnifiedResourceAttributesInfo unifiedResourceAttributesInfo) throws RMException,java.rmi.RemoteException ■ public RMRequestIdInfo updateAttributesBulkByList(RMSessionData rmSessionData,UnifiedResourceBulkAttributeListInfo list) throws RMException,java.rmi.RemoteException
PackageManager	public void invokePackageActivity(AdminData iAdminData, InvokePackageActData pkgActData) throws DataNotFoundException, UrmSystemException, UrmApplicationException, RemoteException;	RM1UnifiedResourceManager	■ public void invokeUnifiedResourceActivity(RMSessionData RMSessionData,UnifiedResourceActivityInfo unifiedResourceActivity) throws RMException, java.rmi.RemoteException ■ public RMInvokeActivityListStatusInfo invokeUnifiedResourceActivityList(RMSessionData rmSessionDataExt, UnifiedResourceBulkActivityListInfo list) throws RMException, RemoteException

Class Name	Method Name	Alternative Class Name	Alternative Method Name
PackageManager	public void createPackage(AdminData adminData, CreatePackageData packageData) throws DataNotFoundException, UrmSystemException, UrmApplicationException, RemoteException;	RM1UnifiedResourceManager	■ public RMInvokeActivityListStatusInfo groupUnifiedResourcesList(RMSessionData session, CreateGroupUnifiedResourcesListBulkInfo list) throws RMException, RemoteException ■ public void groupUnifiedResources(RMSessionData sessionData,NewPackageInfo newPackageData) throws RMException,java.rmi.RemoteException
PackageManager	public QuantityProcessedData invokeBulkPackageActivity(AdminData adminData, InvokeBulkPackageActData bulkpckgActData) throws DataNotFoundException, UrmSystemException, UrmApplicationException, RemoteException;	RM1UnifiedResourceManager	■ public RMRequestIdInfo invokeBulkActivity(RMSessionData RMSessionData, UnifiedResourceBulkActivityInfo bulkActivity) throws RMException, RemoteException ■ public RMRequestIdInfo invokeBulkActivity(RMSessionData RMSessionData, UnifiedResourceBulkActivityListInfo bulkActivity) throws RMException, RemoteException
PackageManager	public QuantityProcessedData createBulkPackages(AdminData adminData, CreateBulkPackagesData bulkPackageData) throws DataNotFoundException, UrmSystemException, UrmApplicationException, RemoteException;	RM1UnifiedResourceManager	■ public RMRequestIdInfo groupUnifiedResourcesBulk(RMSessionData sessionData,GroupUnifiedResourcesSelectionCriteriaBulkInfo selectionCriteria) throws RMException,java.rmi.RemoteException ■ public RMRequestIdInfo groupUnifiedResourcesBulk(RMSessionData sessionData,CreateGroupUnifiedResourcesListBulkInfo bulkList) throws RMException,java.rmi.RemoteException

Class Name	Method Name	Alternative Class Name	Alternative Method Name
PackageManager	public void updatePackageAttributes(AdminData adminData, UpdatePackageAttrData packageAttrData) throws DataNotFoundException, UrmSystemException, UrmApplicationException, RemoteException;	RM1UnifiedResourceManager	■ public void UpdateAttributes(RMSessionData rmSessionData,UnifiedResourceAttributesInfo unifiedResourceAttributesInfo) throws RMException,java.rmi.RemoteException ■ public RMInvokeActivityListStatusInfo updateAttributesList(RMSessionData rmSessionData,UnifiedResourceBulkAttributeListInfo list) throws RMException,java.rmi.RemoteException ■ public RMRequestIdInfo updateAttributeBulk(RMSessionData rmSessionData,UnifiedResourceBulkSelectionCriteriaInfo selectionCriteriaInfo,UnifiedResourceAttributesInfo unifiedResourceAttributesInfo) throws RMException,java.rmi.RemoteException ■ public RMRequestIdInfo updateAttributesBulkByList(RMSessionData rmSessionData,UnifiedResourceBulkAttributeListInfo list) throws RMException,java.rmi.RemoteException
PackageManager	public PackageInStockListData getPackageForActivity(AdminData adminData, InvokeBulkPackageActData bulkpckgActData) throws DataNotFoundException, UrmSystemException, UrmApplicationException, RemoteException;	RM1UnifiedResourceServices	public UnifiedResourceListInfo getUnifiedResourceForActivity(RMSessionData rmSessionData, UnifiedResourceBulkActivityInfo unifiedResourceBulkActivityInfo, PaginationInfo paginationInfo) throws RMException, RemoteException;
ApplManagerConv	public ResourceInStockListData getResourceListInStock (PaginationData pageData, ResourceListData resourceListData) throws DataNotFoundException, UrmSystemException, UrmApplicationException, RemoteException;	RM1UnifiedResourceServices	public UnifiedResourceListInfo getUnifiedResourceList(RMSessionData rmSessionData, UnifiedResourceCriteriaInfo UnifiedResourceCriteriaInfo, PaginationInfo PaginationInfo) throws RMException, RemoteException;

Class Name	Method Name	Alternative Class Name	Alternative Method Name
ApplManagerConv	public PackageInStockListData getPackageList (PaginationData pageData, PackageListData packageListData) throws DataNotFoundException, UrmSystemException, UrmApplicationException, RemoteException;	RM1UnifiedResourceServices	public UnifiedResourceListInfo getUnifiedResourceList(RMSessionData rmSessionData, UnifiedResourceCriteriaInfo UnifiedResourceCriteriaInfo, PaginationInfo PaginationInfo) throws RMException, RemoteException;
ApplManagerConv	public ResourceDetailsData getResourceDetails (AdminData adminData, ResourceValueData resourceValue) throws DataNotFoundException, UrmSystemException, UrmApplicationException, RemoteException;	RM1UnifiedResourceServices	public UnifiedResourceDetailsInfo getUnifiedResourceDetails(RMSessionData RMSessionData, RMEntityIdInfo unifiedEntetyIdInfo) throws RMException, RemoteException;
ApplManagerConv	public boolean checkResourceAvailability (AdminData adminData, IsAvailableResourceData checkAvailabilityData) throws DataNotFoundException, UrmSystemException, UrmApplicationException, RemoteException;	RM1UnifiedResourceServices	public AvailabilityIndicator checkUnifiedResourceAvailability(RMSessionData RMSessionData, UnifiedResourceActivityAvailabilityInfo unifiedResourceActivityAvailabilityInfo) throws RMException, RemoteException;
ApplManagerConv	public boolean checkPackageAvailability (AdminData adminData, IsAvailablePackageData checkAvailabilityData)throws DataNotFoundException, UrmSystemException, UrmApplicationException, RemoteException;	RM1UnifiedResourceServices	public AvailabilityIndicator checkUnifiedResourceAvailability(RMSessionData RMSessionData, UnifiedResourceActivityAvailabilityInfo unifiedResourceActivityAvailabilityInfo) throws RMException, RemoteException;
ApplManagerConv	public ResourceInStockData [] getNextAvailableResource (AdminData adminData, NextAvailableResourceData availableResourceData) throws DataNotFoundException, UrmSystemException, UrmApplicationException, RemoteException;	RM1UnifiedResourceServices	public UnifiedResourceListInfo getNextAvailableUnifiedResourceList(RMSessionData rmSessionData, UnifiedResourceCriteriaInfo unifiedResourceCriteriaInfo, PaginationInfo paginationInfo, UnifiedResourceActivityInfo UnifiedResourceActivityInfo) throws RMException, RemoteException;
ApplManagerConv	public ResourceInStockData [] getNextAvailableResource (AdminData adminData, AvailableResourceCriteriaSetData availbleResourceData) throws DataNotFoundException, UrmSystemException, UrmApplicationException, RemoteException;	RM1UnifiedResourceServices	public UnifiedResourceListInfo getNextAvailableUnifiedResourceList(RMSessionData rmSessionData, UnifiedResourceCriteriaInfo unifiedResourceCriteriaInfo, PaginationInfo paginationInfo, UnifiedResourceActivityInfo UnifiedResourceActivityInfo) throws RMException, RemoteException;

Class Name	Method Name	Alternative Class Name	Alternative Method Name
ApplManagerConv	public PackageDetailData [] getNextAvailablePackage (AdminData adminData, AvailablePackageCriteriaSetData availablePackageData) throws DataNotFoundException, UrmSystemException, UrmApplicationException, RemoteException;	RM1UnifiedResourceServices	public UnifiedResourceListInfo getNextAvailableUnifiedResourceList(RMSessionData rmSessionData, UnifiedResourceCriteriaInfo unifiedResourceCriteriaInfo, PaginationInfo paginationInfo, UnifiedResourceActivityInfo UnifiedResourceActivityInfo) throws RMException, RemoteException;
ApplManagerConv	public PackageDetailData getPackageByComponent(AdminData adminData, PackageIdInputData packageIdInputData) throws DataNotFoundException, UrmSystemException, UrmApplicationException, RemoteException;	RM1UnifiedResourceServices	public UnifiedResourceInfo getUnifiedResourceByComponent(RMSessionData RMSessionData, RMEntityIdInfo unifiedEntetyIdInfo) throws RMException, RemoteException;
	public PackageDetailData getPackageHierarchy(AdminData adminData, PackageIdInputData packageIdInputData) throws DataNotFoundException, UrmSystemException, UrmApplicationException, RemoteException;	RM1UnifiedResourceServices	public UnifiedResourceHierarchyInfo getUnifiedResourceHierarchy(RMSessionData RMSessionData, RMEntityIdInfo unifiedEntetyIdInfo) throws RMException, RemoteException;
	public PackageDetailData getPackageDetails(AdminData adminData, PackageData packageData) throws DataNotFoundException, UrmSystemException, UrmApplicationException, RemoteException;	RM1UnifiedResourceServices	public UnifiedResourceDetailsInfo getUnifiedResourceDetails(RMSessionData RMSessionData, RMEntityIdInfo unifiedEntetyIdInfo) throws RMException, RemoteException;
	public PoolSummaryStatusCountData [] getPoolSummaryDetails(AdminData adminData, PoolSummaryDetailsData poolSummaryDetailsData) throws DataNotFoundException, UrmSystemException, UrmApplicationException, RemoteException;	NA	
	public CategorySummaryDetailsData [] getCategorySummaryDetails(AdminData adminData, CategorySummaryDetailInputData categoryInputData) throws DataNotFoundException, UrmSystemException, UrmApplicationException, RemoteException;	NA	
	public ResourceInStockListData getResourcesByStsAndDate(AdminData adminData, StatusDateData statusDateData) throws DataNotFoundException, UrmSystemException, UrmApplicationException, RemoteException;	NA	

Class Name	Method Name	Alternative Class Name	Alternative Method Name
	public PackageData [] getPackagesByStsAndDate(AdminData adminData, StatusDateData statusDateData) throws DataNotFoundException,UrmSystemException,UrmApplicationException,RemoteException;	NA	
	public PackageHistoryDetailData getPackageHistory (AdminData adminData, PackageData packageData) throws DataNotFoundException,UrmSystemException,UrmApplicationException,RemoteException;	RM1UnifiedResourceServices	public UnifiedResourceInfo[] getUnifiedResourceHistory(RMSessionData RMSessionData, RMEntityIdInfo entityIdInfo) throws RMException, RemoteException;
	public ResourceInStockListData getResourceListByPool(AdminData adminData, PaginationData pagingData, PoolIdData poolId) throws DataNotFoundException, UrmSystemException, UrmApplicationException,RemoteException;	RM1UnifiedResourceServices	public UnifiedResourceListInfo getUnifiedResourceListByPool(RMSessionData rmSessionData, PoolIdInfo poolIdInfo, PaginationInfo paginationInfo) throws RMException, RemoteException;
	public ResourceInStockListData getResourceHistory(AdminData adminData,ResourceValueData resourceValue) throws DataNotFoundException, UrmSystemException, UrmApplicationException, RemoteException;	RM1UnifiedResourceServices	public UnifiedResourceInfo[] getUnifiedResourceHistory(RMSessionData RMSessionData, RMEntityIdInfo entityIdInfo) throws RMException, RemoteException;

Modify Changed APIs

Class Name	Method Name	Type	Removal Reason	Alternative
Utility	All methods	Bean Removal	The parallel functionality has moved to the RM-BE (e.g. the RMA initiates the Bulk request, and the RM-BE is responsible to process it in parallel.)	
ResourceManager	public ResourceValuePartitionIdListData getAffectedResources(AdminData adminData ,InvokeResourceBulkActData rscBulkAct) throws DataNotFoundException,UrmSystemException,UrmApplicationException,RemoteException;	Method Removal	The parallel functionality has moved to the RM-BE (e.g. the RMA initiates the Bulk request, and the RM-BE is responsible to process it in parallel.)	
ResourceManager	public ResourceValueData[] generateResourceList(AdminData adminData ,InvokeResourceBulkActData rscBulkAct) throws UrmApplicationException,UrmSystemException,RemoteException;	Method Removal	The parallel functionality has moved to the RM-BE (e.g. the RMA initiates the Bulk request, and the RM-BE is responsible to process it in parallel.)	

Class Name	Method Name	Type	Removal Reason	Alternative
ResourceManager	public RequestFlowData getRequestFlow(AdminData adminData, InvokeResourceBulkActData rscBulkAct) throws DataNotFoundException, UrmApplicationException, UrmSystemException , RemoteException ;	Method Removal	The parallel functionality has moved to the RM-BE (e.g. the RMA initiates the Bulk request, and the RM-BE is responsible to process it in parallel.)	
PackageManager	public ResourceValuePartitionIdListData getAffectedPackages(AdminData adminData, InvokeBulkPackageActData bulkpckgActData) throws DataNotFoundException, UrmSystemException, UrmApplicationException, RemoteException;	Method Removal	The parallel functionality has moved to the RM-BE (e.g. the RMA initiates the Bulk request, and the RM-BE is responsible to process it in parallel.)	
PackageManager	public BulkPackageComponentsData [] getBulkPackageComponents(AdminData adminData, CreateBulkPackagesData bulkPackageData, byte mode) throws DataNotFoundException, UrmSystemException, UrmApplicationException, RemoteException;	Method Removal	The parallel functionality has moved to the RM-BE (e.g. the RMA initiates the Bulk request, and the RM-BE is responsible to process it in parallel.)	
PackageManager	public QuantityProcessedData createBulkPackagesParallel(AdminData adminData, CreatePackageData [] crPackData) throws DataNotFoundException, UrmSystemException, UrmApplicationException, RemoteException;	Method Removal	The parallel functionality has moved to the RM-BE (e.g. the RMA initiates the Bulk request, and the RM-BE is responsible to process it in parallel.)	
PackageManager	public QuantityProcessedData invokeBulkPackageActivityParallel(AdminData adminData, InvokePackageActData[] packActvData) throws DataNotFoundException, UrmSystemException, UrmApplicationException, RemoteException;	Method Removal	The parallel functionality has moved to the RM-BE (e.g. the RMA initiates the Bulk request, and the RM-BE is responsible to process it in parallel.)	
ApplManagerConv	public ResourceValueData getResourceMaxValue(AdminData adminData, ResourceMaxValueData resourceMaxValue) throws DataNotFoundException, UrmSystemException, UrmApplicationException, RemoteException;	Method Removal		
	public ResourceValueData [] allocateFreeResources (AdminData adminData, AllocateFreeResourcesData allocateFreeResourcesData) throws DataNotFoundException,UrmSystemException,UrmApplicationException,RemoteException;	Method Removal	The parallel functionality has moved to the RM-BE (e.g. the RMA initiates the Bulk request, and the RM-BE is responsible to process it in parallel.)	

Class Name	Method Name	Type	Removal Reason	Alternative
	public TableSeqData getCurrentTableSeq(AdminData adminData, String tableSeqName, int range) throws DataNotFoundException, UrmSystemException, UrmApplicationException, RemoteException;	Method Removal	The RM-BE utilizes the Oracle Sequences in order to generate unique ID's, and the parallel functionality has moved to the RM-BE (e.g. the RMA initiates the Bulk request, and the RM-BE is responsible to process it in parallel.)	

Integrate with SONAR

Modify customization configuration files to use the SONAR standard.

Adjust Batch Processes

The following batch processes where removed:

- Age Out
- Release
- Unlock

The above batch processes were replaced with one daemon process (see the following section)

In order to develop a new batch process, or customize an existing one, please refer to the RM Customization document: ABP - RM 6.0 - Resource Management Customization Guide #189110

Adjust Daemon Processes

The following daemon process is a new one:

- Request and End State Daemon. For more details regarding this new daemon process please refer to the following document: *ABP - RM 6.0 – Resource management Run Bock #186606.*

In order to develop a new Daemon process, or customize an existing one, please refer to the RM Customization document: ABP - RM 6.0 - Resource Management Customization Guide #189110

Update Configuration Changes

This section details all configuration changes that were made in the core layer that impact the customization layer. Configuration properties are maintained in properties files and other various configuration files.

Properties Files

All TopLink properties files were removed.

Other Configuration Files Changes

All TopLink configuration files were removed.

Update Customization Interface

Customization Public Objects

The Resource Management version 6.0 is integrated with the ABP Java Foundation; the Java Foundation introduces a full Customization Framework, for detailed expatiation regarding the new Customization framework, please refer to the document: ABP - RM 6.0 - Resource Management Customization Guide #189110.

The following non-flow handlers are deprecated:

- amdocs.rm3g.customizationexits.handlers.PackageHandler
- amdocs.rm3g.customizationexits.handlers.ResourceHandler

The following non-flow handler replaces the above two:

- amdocs.rm3g.customizationexits.nonFlow.handlers UnifiedResourceCustomization

Backward Compatibility

In order to be able to work against the deprecated handlers, the customized handler implementation must inherit from the following classes:

- amdocs.rm3g.customizationexits.handlers.AbstractPackageType
- amdocs.rm3g.customizationexits.handlers.AbstractResourceType

Removed External Imports:

In addition, the following imports were removed:

- All Toplink imports

Removed Core Packages:

The following core packages were removed:

- amdocs.rm3g.accesscontrol.*
- amdocs.rm3g.accesscontrol.ldap.*
- amdocs.rm3g.accesscontrol.sum.*
- amdocs.rm3g.messages.*
- amdocs.rm3g.rmtlmodel.*
- amdocs.rm3g.rmtlmodel.deployment.*
- amdocs.rm3g.rmtlmodel.project.*
- amdocs.rm3g.urm.datalayer.*
- amdocs.rm3g.urm.datamodel.*
- amdocs.rm3g.urm.dbinfra.*
- amdocs.rm3g.urm.jobs.*
- amdocs.rm3g.urm.processlayer.*

Any referral to any class from the above packages must be removed, otherwise the customized classes will not compile.

Automatic Upgrade Scripts

Script Location and Name	Description	Execution Prerequisites
~ccip/bb/gurm/v60_0/upgrade_scripts/src/ABP_UPG_RM_V550_V600_RM1_CATEGORY_ASSIGNMENTS.sql	Populates the new reference table RM1 CATEGORY ASSIGNMENTS with all the combination per resource type, category type and values. For more details regarding this table, please refer to the data model changes section.	Non
~ccip/bb/gurm/v60_0/upgrade_scripts/src/ABP_UPG_RM_V550_V600_RM_LIFE_CYCLE_ACT_CONV_STS_UN_PACKED.sql	Converts all the UN_PACKED status to NOT_EXIST in order to be physically deleted, since the decision to physically delete a package / resource from the database is made only if the target status is NOT_EXIST	Non
~ccip/bb/gurm/v60_0/upgrade_scripts/src/ABP_UPG_RM_V550_V600_RM_PARAM_TYPE.sql	Converts the core parameter type implementation classes to the new ones	Non
~ccip/bb/gurm/v60_0/upgrade_scripts/src/ABP_UPG_RM_V550_V600_RM_RULE_TYPE.sql	Converts the core rule type implementation classes to the new ones	Non

Tools (RTS)

Summary of Upgrade Activities

The following table summarizes the activities that participate in the development upgrade process. The activities are presented in the recommended order of their execution. Further details about each activity are described in subsequent sections.

Activity	Description	Scheduling	Automatic Process	Prerequisite for Development	Comments
Stop Using Deprecated APIs	Several APIs are declared as deprecated, and may be removed in the next release. This step is optional, if it is required to stop using these APIs, and start using the alternatives	Development	No	No	Performing this step will save the need to handle the "deprecated APIs removal" step in the next version
Integrate with SONAR		Development	No	No	
Update XML API XSDs	Generate new APIs	Development	Yes	No	

Stop Using Deprecated APIs

The following APIs were declared as deprecated in this version and should not be used in the long term. As deprecated APIs can be removed from future versions, it is highly recommended to stop using these APIs. The Deprecated view in Javadoc HTML documentation lists all the deprecated methods.

Class Name	Method Name	Type	Alternative
amdocs.acmarch.invoker.executer.Invoker	execute	String xmlContent Connection aFlConn String aApplicationCode	No
amdocs.acmarch.invoker.executer.Invoker	exceute	String xmlContent Object[] constructorParams Class[] constructorClasses Connection aFlConn	Use the local session API

All the A&C Java APIs which were JNI-based are now deprecated and a new set of A&C Java API exists. For more detailed please see the A&C User Guide.

Modify Changed APIs

APInvoker Executer has new API:

```
public Object exceute(String xmlContent, Object aObject,
Connection aFlConn) throws ACMException
```

Integrate with SONAR

Modify customization configuration files to use the SONAR standard.

In this new version APInvoker will automatically load all application that is defined in the sonar environment, so there's no need anymore to define in op_trb1apinvoker_env_sh the names of each component.

Update XML API XSDs

Two new fields were added to the TRB XSDs in order to handle the transactions dependency:

- Entity type – customer / subscriber
- Entity ID.

The new XSDS are generated automatically in the CC by the XSDGenerator.

The user needs to make sure that these XSD exists and that they include the new fields.

Transaction Broker

Summary of Upgrade Activities

The following table summarizes the activities that participate in the development upgrade process. The activities are presented in the recommended order of their execution. Further details about each activity are described in subsequent sections

#	Activity	Description	Scheduling	Automatic Process	Prerequisite for Development	Comments
1		Database changes are described in the Transaction Broker Release Notes. The automatic database upgrade process is responsible for the structure and data upgrade.	Development	Yes	Yes	
2	Remove APIs	Adjust any code that uses a deprecated API that has been removed in the new version	Pre build	No	Yes	Running the build process is a simple way to find these removed APIs
3	Update Operational Changes	To support Multi Engine environment, the operational tables have been updated. In addition, the update was done to support the integration with AMC team.	Development	Yes	Yes	
4	Update XML API XSDs	The Registration file of an application to the Transaction Broker has been updated. In addition, each publishing application should update its transaction xml format to support new attributes.	Development	Yes	No	

Remove APIs

The following deprecated objects/methods were removed from this version.

In order to adjust the customized code to this change, it is required to stop using these objects and methods.

Class Name	Method Name	Type	Removal Reason	Alternative
TRBSubscriberMessage	void TRBSubscriberMessage:: TRBSetError(TRBApplicationErrorCodeType Error, char* ErrorDesc)	Method removal	There is no use in Error Type anymore	In the new Error Handling mechanism the Errors are grouped in Categories
TRBSubscriberMessage	TRBErrCodeType TRBSubscriber:: TRBDeleteMessages(long FromTrxSeqNo, long ToTrxSeqNo)	Method removal	There is no more deletion by the API from TRB1_SUB_LOG.	The cleaning of the table will be done by the Clean Up process.
DLSubscriberLog	public void delSubLogBySeq(BigDecimal aApplId, BigDecimal aTrxId) public void delSubLogBySeq(BigDecimal aApplId, BigDecimal aFromSeqNo, BigDecimal aToSeqNo) public void delSubLogByReq(BigDecimal aApplId, BigDecimal aFromSeqNo, BigDecimal aToSubTrx)	Methods removal	There is no more deletion by the API from TRB1_SUB_LOG.	The cleaning of the table will be done by the Clean Up process.

Update Operational Changes

Tables GN1_TASK_CONNECT and GN1_CONNECT_PARAMS are updated with new values:

- Parameters to support the Multi Engine Framework. These parameters define the port that each engine locks when it runs.
- Parameters for the support of the AMC in multi engine environment.

Update XML API XSDs

- The TRB_FL_TRB_TRB_REGISTER.xsd file has been updated with the attribute MultiThreadInd. Need to update the UTL1_FILE_LOADER table with the new XSD file.
- Each publishing application should update the format of the transactions XML file that it sends, according to the new format. The new format is added with the attributes: EntityType and EntityId.

Voucher Management

Summary of Upgrade Activities

The following table summarizes the activities that participate in the development upgrade process. The activities are presented in the recommended order of their execution. Further details about each activity are described in subsequent sections.

#	Activity	Description	Scheduling	Automatic Process	Prerequisite for Development	Comments
1	Regenerate Valid Values	Delete all old core domains in the customization area, add all domains VVs via Fox	Development	No	No	The old solution is applicable, but not recommended
2	Adjust Application Make Files	Adjust the customized make files to the new version	Pre Build	No	No	
3	Regenerate Domains	Generate all customized domains	Build	Yes	Yes	
4	Regenerate Data Types	Rebuild the customization area	Build	Yes	Yes	
5	Change DVM files format	Run DVM migration script	Development	No	Yes	The DVM is now part of the dStudio, the format change is mandatory to comply with the studio standard. See Appendix B: DVM, RDM Persistence Tools - Migration Script
6	Regenerate Database Views (JUTIL 5.2 Packages)	Regenerate all customized views	Pre Build	Semi (need to check in files manually)	Yes	
7	Migrate Reference datalayers to DVM	Run DVM migration script	Development	No	Yes	. See Appendix B: DVM, RDM Persistence Tools - Migration Script
8	Regenerate Reference Data Layers (JUTIL 5.2 Packages)	Regenerate all customized data layers	Pre Build	Semi (need to check in files manually)	Yes	
9	Update Message Handling Services (JUTIL 5.2 Packages)	Check if code changes are needed due to interface change	Pre Build	No	Yes	

#	Activity	Description	Scheduling	Automatic Process	Prerequisite for Development	Comments
10	Modify Changed Database Structures					
11	Remove APIs	Adjust any code that uses a deprecated API that has been removed in the new version	Pre build	No	Yes	Running the build process is a simple way to find these removed APIs
12	Stop Using Deprecated APIs	Several APIs are declared as deprecated, and may be removed in the next release. This step is optional, if it is required to stop using these APIs, and start using the alternatives	Development	No	No	Performing this step will save the need to handle the "deprecated APIs removal" step in the next version
13	Update Operational Changes	See details in subsequent sections.				
14	Operational Jobs	Modify customization configuration files to use the SONAR standard	Development	No	No	
15	Adjust Batch Processes	Integration with SONAR. Adjust the modified core's scripts Operational setup	Development	No	Yes	
16	Update Configuration Changes	See details in subsequent sections.				
17	Automatic Upgrade Scripts	Script to run in order to perform the upgrade in automatic manner	Defined by the script's context	Yes	Defined by the script's context	Scripts that replace the manual upgrade activities, e.g. data migration, for XML's, DB, operational, shell scripts etc.

Regenerate Valid Values

All customized Fox objects need to be regenerated prior to performing the upgrade.

Adjust Application Make Files

The following table summarizes the customized Make file changes that are relevant for version 6.0.

Item	Change Type	Description	Required Steps
SONAR template builder	New	Generator which builds SONAR template files from input application properties files	Add it to your Make file. Make sure that your properties comply with the SONAR tags standard

Regenerate Domains

All customized domain objects need to be regenerated prior to performing the upgrade.

Regenerate Database Views (JUTIL 5.2 Packages)

All customized DVM views need to be upgraded and regenerated prior to performing the upgrade. See Appendix B: DVM, RDM Persistence Tools - Migration Script

Regenerate Reference Data Layers (JUTIL 5.2 Packages)

All RDM customization datalayers need to be migrated and regenerated prior to performing the upgrade. See Appendix B: DVM, RDM Persistence Tools - Migration Script.

Modify Changed Database Structures

Database changes are described in the Voucher Management Release Notes. The automatic database upgrade process is responsible for the structure and data upgrade. The data upgrade is required when there is a need to upgrade a live environment containing data from 5.5 to 6.0.

The Tiger tool supports running the upgrade.

As database objects may be used by customizations (e.g., map customization views to core table fields), and removed or modified objects may impact the customization layer, **these changes are listed here in addition to the Release Notes.**

DB Object Area & GDD BB	Object Name	Object Type	Object Change Type	Column Name	Column Change Type	Domain	Change Description
gvmgdd	VM1_ORDER	Table	dropped	RSN_CD			
gvmgdd	VM1_ORDER	Table	dropped	REMARK			
gvmgdd	VM1_ORDER_HISTORY	Table	dropped	RSN_CD			
gvmgdd	VM1_ORDER_HISTORY	Table	dropped	REMARK			
gvmgdd	VM1_PACKAGE_HISTORY	Table	dropped	RSN_CD			
gvmgdd	VM1_PACKAGE_HISTORY	Table	dropped	REMARK			
gvmgdd	VM1_PACKAGE	Table	dropped	RSN_CD			
gvmgdd	VM1_PACKAGE	Table	dropped	REMARK			
gvmgdd	VM1_SEQ_ARRAY	Table	New				Sequence Management Table
gvmgdd	VM1_SEQ_ARRAY.IDAT	Table	New				Initial Data
gvmgdd	VM1_FIELD_DEST	Ref Table	New	MAPPING_CLASS		VARCHAR2(50)	New column
gvmgdd	VM1_FIELD_DEST	Ref Table	New	MAPPING_FIELD		VARCHAR2(50)	New column
gvmgdd	VM1_FIELD_DEST	Ref Table	Dropped	DEST_TABLE_NAME			
gvmgdd	VM1_FIELD_DEST	Ref Table	Dropped	DEST_FIELD_NAME			

Remove APIs

The following deprecated objects/methods were removed from this version.

In order to adjust the customized code to this change, it is required to stop using these objects and methods.

Class Name	Method Name	Type	Alternative
VM1VoucherManager	getFileActivity	Remove	N/A
	getFileActivities	Remove	Class: SearchServiceBean Method: searchFileActivity
	getVoucher	Remove	Class: SearchServiceBean Method: serachVoucher
	getVoucherWithHistory	Remove	Class: SearchServiceBean Method: getVoucherHistory
	getVoucherHistory	Remove	Class: SearchServiceBean Method: getVoucherHistory
	getVouchers	Remove	N/A
	getVouchers	Remove	Class: SearchServiceBean Method: serachVoucher
	getNextVouchers	Remove	N/A
	GetPreviousVouchers	Remove	N/A
	HasNextVouchers	Remove	N/A
	hasPreviousVouchers	Remove	
	changeVocherState	Remove	Class: VoucherServiceBean Method: updateVoucher
	changesVoucherState	Remove	Class: VoucherServiceBean Method: updateVouchers
VM1VoucherRecharge	voucherRecharge	Remove	Class: VoucherServiceBean Method: voucherReachrge
	voucherReachargeByPin	Remove	Class: VoucherServiceBean Method: voucherReachrge
	cancelVoucherRecharge	Remove	Class: VoucherServiceBean Method: cancelVoucherRecharge
VM1VoucherOrderManagement	selectOrder	Remove	N/A
	selectOrders	Remove	Class: SearchServiceBean Method: searchOrdrer
	getNextOrderBulk	Remove	N/A
	getPreviousOrderBulk	Remove	N/A
	hasNextOrderBulk	Remove	N/A
	hasPreviousOrderBulk	Remove	N/A
	selectOrdersTreatment	Remove	Class: SearchServiceBean Method: searchOrderTraetment
	selectPackage	Remove	N/A
	selectPackages	Remove	Class: SearchServiceBean Method: searchPackage
	getNextPackageBulk	Remove	N/A

Class Name	Method Name	Type	Alternative
	getPreviousPackageBulk	Remove	N/A
	hasNextPackageBulk	Remove	N/A
	hasPreviousPackageBulk	Remove	N/A
	selectPackagesTreatment	Remove	Class: SearchServiceBean Method: searchPackageTreatment
	getNextPackageTreatmentBulk	Remove	N/A
	TreatmentgetPreviousPackageTreatmentBulk	Remove	N/A
	hasNextPackageTreatmentBulk	Remove	N/A
	hasPreviousPackageTreatmentBulk	Remove	N/A
	packageSentToDealer	Remove	Class: PackageServiceBean Method: packageSentToDealer
	packagesSentToDealer	Remove	Class: PackageServiceBean Method: packagesSentToDeale
	packageReceivedByDealer	Remove	Class: PackageServiceBean Method: packagesReceivedByDealer
	packagesReceivedByDealer	Remove	Class: PackageServiceBean Method: packagesReceivedByDealer
	stolenPackage	Remove	Class: PackageServiceBean Method: stolenPackage
	stolenPackages	Remove	Class: PackageServiceBean Method: stolenPackages
	createOrder	Remove	Class: PackageServiceBean Method: createOrder
	checkOrder	Remove	Class: PackageServiceBean Method: checkOrder
	updateOrder	Remove	Class: PackageServiceBean Method: updateOrder
	authorizeOrder	Remove	Class: PackageServiceBean Method: authorizeOrder
	sendOrder	Remove	Class: PackageServiceBean Method: sendOrder

A simple build process easily generates a list of all areas in the customized code that need to be fixed.

Stop Using Deprecated APIs

N/A

Update Operational Changes

Operational Jobs

The following table summarizes all the Operational jobs that are new or changed in this version:

Functionality	Job Name	Script	Change Type	Dependency	Description
VM Refresh Job	VM1REFRESH	Vm1Refresh_Sh	Deleted	None	Replaced by the 24X7 EJ refresh agent.

Operational Parameter Tables

N/A

Job Dependencies

N/A

See the VM Run Book for more details regarding this job's parameter.

Adjust Batch Processes

Batch processes now has customization exit points, both standard DVM (database related) and business where applicable.

Update Configuration Changes

This section details all configuration changes that were made in the core layer that impact the customization layer. Configuration properties are maintained in properties files and other various configuration files.

Properties Files

Properties files from older versions were replaced with new properties files according to the Java Foundation (JF) standards.

The new properties files are:

- VM1App.proeprties – contains all VM application properties for both EJB and batch modes.
- VM1AppEJB – Contains VM connection data
- VM1Env –
- VM1GeneralLogHandler – Contains VM log files properties.
- VM1AppBatch.proeprties –contains VM batch specific properties.
- VM1JF.proeprties – contains standard JF properties.
- VM1JFBatch.proeprties – contains standard JF properties for batch jobs and daemons.

- VM1JFEJB.proeprties – contains standard JF properties for EJB mode.

See the "Properties/Implementation Changes" section in the Release Notes document.

In addition see SONAR changes section in this document.

Automatic Upgrade Scripts

No special upgrade script is required.

Script Location and Name	Execution Prerequisites
ABP_UPG_< application ID as in the CC > V550_V600_<Unique string>.*	

5. UPGRADING THE DATABASE

This chapter describes the databases upgrade and the post installation tasks.

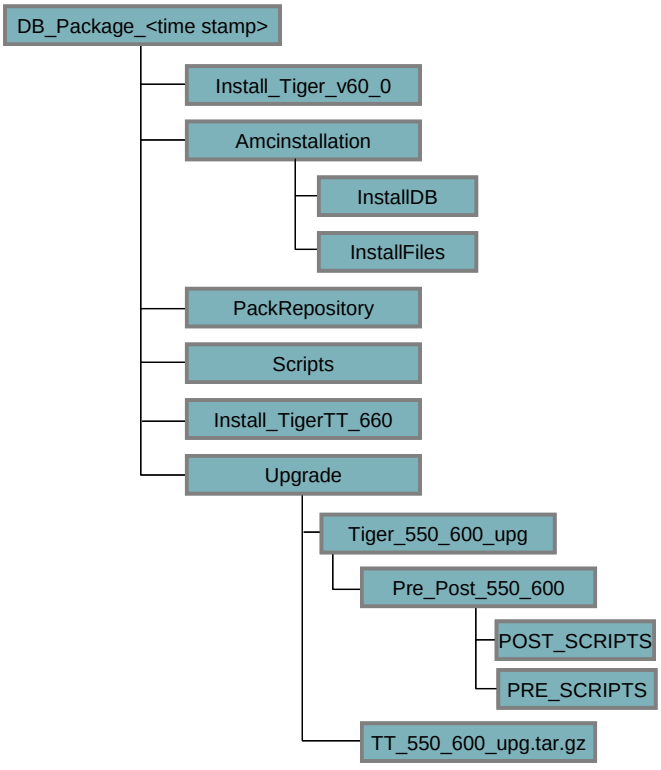
Re-running CC Build

After you completed the minimal requirements for the LEL layer and updated the development environment, you should re-build the CC builds as describe in the Server Installation Guide (CDE), chapter 7.

Section Name	Follow CDE Installation document Yes/No	Description
Running CC Build	Yes	
Checking the Build results.	Yes	

Opening the DB Delivery Content

1. Login into the Tiger UNIX account dbtgr.
2. Copy the Tiger_Installation_v600.tar.gz file from the delivery content into the $HOME directory.
3. Expand the content delivery file:
   `tar -xvf Tiger_Installation_v600.tar.gz`
   The following directories will be created under the home directory:

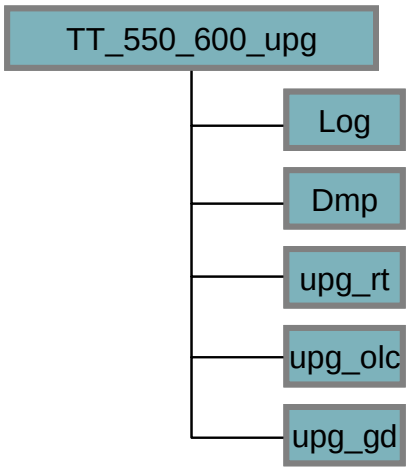


To open the TimesTen Delivery Contents:

From the command line:

1. Log into the timesten Tiger UNIX account.
2. Copy the TT_550_600_upg.tar.gz from the Delivery Contents (DB_Package_<timestamp> -> Upgrade) into the $HOME directory.
3. Extract the TimesTen Delivery Contents file:
   `tar -xvf TT_550_600_upg.tar.gz`

   The following directories are created under home directory:



Upgrade Tiger Software

Dependency: Install Oracle

There are two options to upgrade the Tiger:

1. Upgrade the existing Tiger installation. This requires upgrading the existing instance from 9.2.0.3 to 9.2.0.5.
2. Copying the existing installation to a new location and creating a new Tiger installation based on the existing Tiger repository.

Upgrading an Existing Tiger from Version 5.5 to Version 6.0

Prior to upgrading Tiger to version 6.0, you need to upgrade the AMC to version 6.0.

Upgrading the Amc to Version 6.0

1. Login into Tiger UNIX account dbtgr.
2. Save the Amc directory as a backup. Directory Amc-{host} at the home of Tiger UNIX account.

3. Edit the following environments variables in .profile file:

```
export ORACLE_SID=          # Apply the Oracle instance name
                            #   (e.g. ORACLE_SID=AIMINST9I)
export ORACLE_HOME=         # Apply Oracle installation home directory
                            #   (e.g. ORACLE_HOME=/oravl01/oracle/9.2.0.5)
export JAVA_HOME=           # Apply Java installation home directory
                            #   (e.g. JAVA_HOME=/usr/java1.4.2_06)
export NLS_LANG=            # Apply DB carachter set
            # (e.g. NLS_LANG=AMERICAN_AMERICA.AL32UTF8)

# Set up Installation root directory
export installdir=$HOME/ DB_Package_<timestamp>
```

4. Run Amc Stop.
5. Change directory to $install_dir/AmcInstallation/InstallDB
   `cd $install_dir/AmcInstallation/InstallDB`
6. Connect to the Tiger repository user, e.g. <tiger_username>/<tiger_password>@<db_instance> using SQL and run:
   ```
   drop table amc_conf_repository ;
   drop table amc_history_table ;
   ```
7. Run:
   `@create_all.sql`

This creates five database tables used by AMC

AMC_CONF_REPOSITORY

AMC_HISTORY_TABLE

AMC1_API_EVENTS

AMC1_API_TRAN

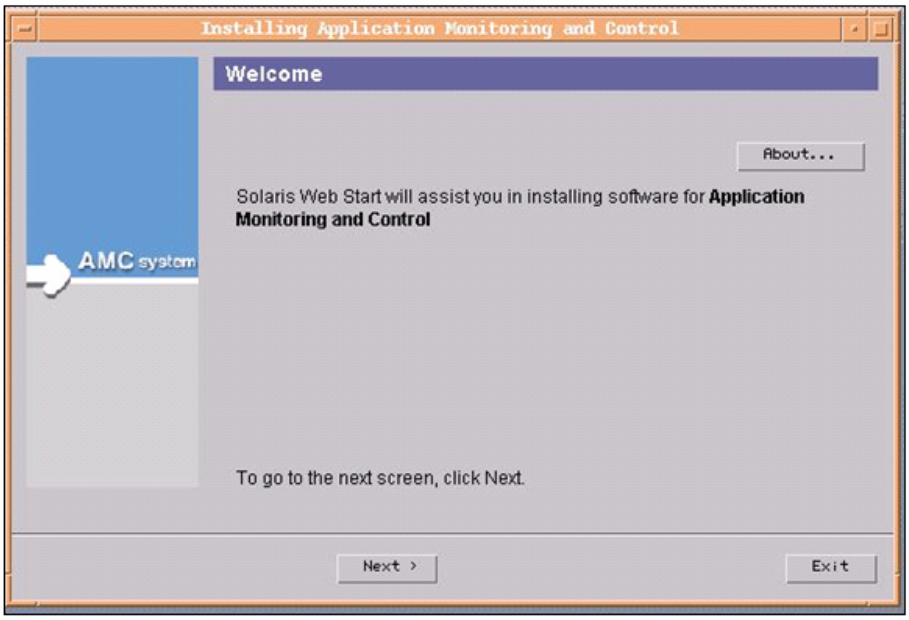
AMC1_API_TRAN_CELLS

8. Define the DISPLAY environment variable with your display IP address:
   export DISPLAY=<IP_Address>
   where <IP_Address> is the IP address of your computer.

You can find your computer's IP address by running the ipconfig command in the Windows Command window, for example:

This is needed for displaying the AMC GUI.

9. Change directory to $install_dir/AmcInstallation/InstallFiles
   `cd ../ InstallFiles`
10. Run:
    `AmcInstaller.ksh`
    and wait for the GUI to open.



11. Click **Next**.
12. In the Setting AMC Environment Variables screen, define the following parameters:

    note *This step includes selecting the port to be used by Tiger for access from your browser. Refer to Appendix I, Port Allocation, in the Server Installation Guide (CDE) for information about port allocation. Be sure to note the port you allocate.*

    - Host – Server on which the AMC will be installed.
    - DB Host – Host on which the AMC database account is located.
    - User Name – Amdocs Billing Platform installation database user name. For a Tiger installation, enter <Tiger Account Name>.
    - Password – Amdocs Billing Platform installation database password. For a Tiger installation, enter <Tiger Account Name>.
    - Instance – Amdocs Billing Platform installation instance.
    - Listener port – Database listener port.

      note *Normally the default is sufficient.*

    - AMC port – Use the port allocated for the AMC in v5.5
13. Select the Custom type of installation. Click **Next.** In the Select Type of Install screen,
14. In the Locale Selection screen click **Next**.

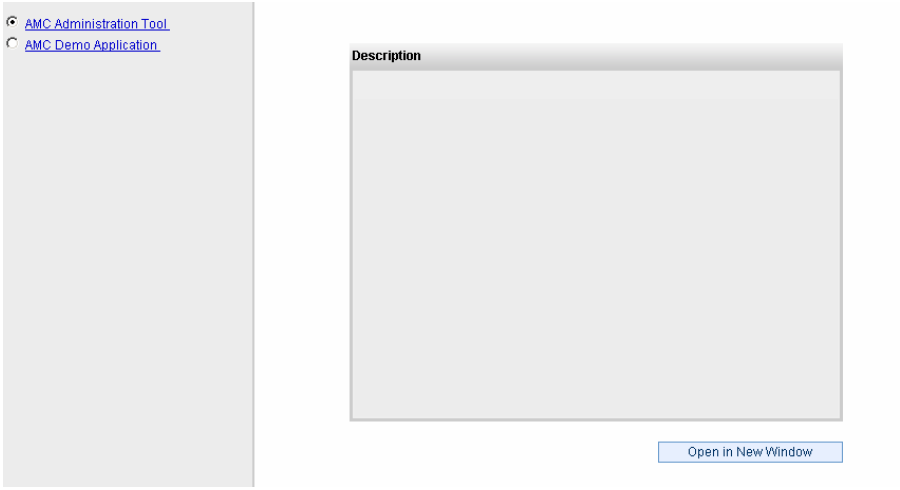
15. In the Select Install Directory screen, modify the installation directory:
    From: <tiger user full path>/Aim_Install/Amc-LocalHost
    To: <tiger user full path>/Amc-LocalHost
16. Confirm the creation of the new install directory by clicking **Yes**.
17. In the Installation Summary screen click **Next**.
18. In the Ready to Install screen click **Install Now**. This action summarizes the installation data.
19. In the Installation Mode Selection screen, select the Master installation, and click **Next**.
    The next screen displays the status of the installation being carried out.
20. In the Post Installation Panel screen, click **Next** when the button becomes active.
21. To view the installation report, click **Details**.
22. Click **Exit.**
23. Change directory to the Home directory:
    `cd $HOME`
24. Run .profile to initialize AMC definitions:
    `../.profile`
25. Start AMC by entering the command:
    `Amc Start`

There are three UNIX commands associated with AMC:
Amc Start – Starts the AMC
Amc Stop – Terminates the AMC
Amc Restart – Used to restart AMC after a server crash

26. Log into the web AMC front-end, using Internet Explorer with the link:
    http://<host>:<amc_port>
27. For initial login, there are some built-in users and passwords:

User	Password	
amcAdmin	Admin11	Admin user

28. Login as amcAdmin. The following window appears.



29. Click **AMC Administration Tool** to enter AMC.
30. To restore the Tiger users of the previous installation, run the AmcUamsDBConverter from Tiger UNIX ~dbtgr/**Amc{host}**/bin as follows:
    `AmcUamsDBConverter amcAdmin/Admin11`

Upgrading Tiger Software and Repository

To upgrade the Tiger software and to upgrade Tiger repository:

1. Change directory to $install_dir/InstallTiger_v60_0
   `cd $install_dir/InstallTiger_v60_0`
2. Run:
   ```
   ./InstallTiger –option 3 –amcdir <../Amc-host>
    – tigerdb <tgrusr/tgrpass@tgrinst> –datatbs
   <tablespace_name> –ixtbs <indexspace_name>
   ```
   where:

Parameter	Description
option 3	Upgrade of all the Tiger files and upgrade of Tiger repository
-amc dir	The full pathname of the AMC for Tiger installation directory.
-tigerdb	Database for Tiger control tables.
-datatbs	Data tablespace. e.g. POOL_DATA
-ixtbs	Index tablespace.e.g. POOL_IX

   For example:
   ```
   ./InstallTiger -option 3 -amcdir
   /dlhuser.p711/inf/aimsys/dlhtgr6/Amc-hpp711/  -tigerdb
   tiger_rep/tiger_rep@STHH9I -datatbs POOL_DATA -ixtbs
   POOL_IX
   ```
3. Wait until you receive confirmation that the command has finished and the following message appears.

```
============================================================
InstallTiger: Finished successfully!
============================================================
```

Creating a New Tiger Installation

1. Create a new oracle account on the 9.2.0.5 repository.
2. Copy Tiger tables from the Tiger repository v5.5 using Oracle export/import utility.
3. Drop AMC tables and sequences (all objects that start with AMC_%)
4. Create a new Tiger UNIX account dbtgr600 and login into it.
5. Install AMC v6.0. Refer to the Server installation Guide (CDE).
6. Change directory to $install_dir/InstallTiger_v60_0
   ```
   cd $install_dir/InstallTiger_v60_0
   ```
7. Run:
   ```
   ./InstallTiger -option 3 -amcdir <../Amc-host>
    - tigerdb <tgrusr/tgrpass@tgrinst> -datatbs
   <tablespace_name> -ixtbs <indexspace_name>
   ```
   where:

Parameter	Description
option 3	Upgrade of all the Tiger files (install if do not exist) and upgrade Tiger repository
-amc dir	The full pathname of the AMC for Tiger installation directory.
-tigerdb	Database for Tiger control tables.
-datatbs	Data tablespace. e.g. POOL_DATA
-ixtbs	Index tablespace.e.g. POOL_IX

   For example:
   ```
   ./InstallTiger -option 3 -amcdir
   /dlhuser.p711/inf/aimsys/dlhtgr6/Amc-hpp711/  -tigerdb
   tiger_rep/tiger_rep@STHH9I -datatbs POOL_DATA -ixtbs
   POOL_IX
   ```
8. Wait until you receive confirmation that the command has finished and the following message appears.

   ```
   ================================================
   InstallTiger: Finished successfully!
   ================================================
   ```

Creating the DB Upgrade Kit

The *Amdocs Billing 5.5 to 6.0 Database Upgrade Kit* provides a set of documents, scripts and files that enable each Amdocs account to upgrade the version 5.5 database structures and data to version 6.0. Any additional changes made during customization development by an account's development team should be packaged and added to this DB Upgrade Kit.

This section describes how to add these additional changes to the Upgrade Kit.

The information in this section is only relevant if the ABP core product has been customized by an account. If no database (oracle + TimesTen) customizations have been performed, then this section can be skipped.

Editing Scripts to Upgrade TimesTen

This upgrade refers to the APDO CC of version 5.5 and version 6.0. Since the scripts include references to the actual *.ttsql files, any changes made in TimesTen tables in the customization layer must be manually implemented in the upgrade scripts by the ADBA on site.

The upgrades scripts are part of the TimesTen TAR file described below, and are simple KSH files that can be edited if necessary.

It is assumed that the object structures in the Amdocs Billing Platform 5.5 data stores are the same as those defined in the Amdocs Billing Platform 5.5 CC.

TimesTen 5120 64-bit must be installed for Amdocs Billing Platform 6.0 (this is the System UNIX team's responsibility).

The upgrade scripts connect to the data stores in direct mode.

Set your environment to point to the correct TimesTen path and version using the TimesTen setting script located in:

${TT_Installation_PATH}/demo/ ttSetEnv.{csh/sh}

This section describes two possible configurations:

- Upgrading two data stores:
  - Guiding
  - Rating and Online Charging
- Upgrading one data store that combines Guiding, Rating and Online Charging

Verifying the Existence of Data Stores

Amdocs Billing Platform 5.5 used TimesTen version 5028, while version 6.0 uses TimesTen version 5120.They both use 64-bit data stores.

Both versions of the Amdocs Billing Platform use a direct configuration (64-bit data store, direct connection). Since the TimesTen versions are different, you must create TimesTen 5120 data stores and copy the old version 5028 data stores into them, as described later in this document.

In Amdocs Billing Platform 6.0, environments have been simplified. This means that one data store is used for Guiding, Rating and Online Charging.

To verify the existence of the data stores:

1. Set your environment to point to the correct TimesTen path and version, using the TimesTen setting script located in:
   - KSH account:

     . /opt/TimesTen/tt5028_64/demo/ttSetEnv.sh
   - TCSH account:

     source /opt/TimesTen/tt5028_64/demo/ttSetEnv.csh

2. Repeat the following two steps for each data store (Guiding and Rating): Use the following this'll command to connect to the data store and verify its existence:
   `ttIsql -connStr dsn=<dsn>`
3. After you have connected to the data store, run the tables command to display the tables:
   `Command> tables;`

Editing the Upgrade Scripts

This procedure refers to the Billing CC of version 5.5 and the APDO CC of version 6.0. Since the scripts include references to the actual *.ttsql files, any changes made in TimesTen tables during customization must be manually implemented in the upgrade scripts by the ADBA at the site.

To edit the upgrade scripts:

1. Manually compare the version 5.5 CC and the version 6.0 CC *.ttsql scripts. The Amdocs Billing Platform 5.5 scripts are located in:
   - Guiding:
     ~<cc_user>/proj/gdgdd550V64/gd_generated/*.ttsql
   - Online Charging:
     ~<cc_user>/proj/olcgdd550V64/olc_generated/*.ttsql
   - Rating, Pricing Engine:
     ~<cc_user>/proj/pmgdd550V64/pm_generated/*.ttsql
   - Rating, Rater:
     ~<cc_user>/proj/rprgdd550V64/rpr_generated/*.ttsql

   The Amdocs Billing Platform 6.0 scripts are located in:
   - Guiding:
     ~<cc_user>/proj/gdgdd600V64/gd_generated/*.ttsql
   - Online Charging:
     ~<cc_user>/proj/olcgdd600V64/olc_generated/*.ttsql
   - Rating, Pricing Engine:
     ~<cc_user>/proj/pmgdd600V64/pm_generated/*.ttsql
   - Rating, Rater:
     ~<cc_user>/proj/rprgdd600V64/rpr_generated/*.ttsql
2. In the UNIX timesten account, create the following directory:
   `~timesten/TimesTen/600/ upg_550_to_600.`
3. Change into this directory (run cd).

4. In the directory, run gunzip and extract (untar) the provided tt1_550_600_upg_scripts.tar.gz file:
```
>gunzip tt1_550_600_upg_scripts.tar.gz
>tar -xvf tt1_550_600_upg_scripts.tar
```
The following directories are created:
   - upg_gd – Contains the GD_550_to_600 script, which recreates Guiding tables.
   - upg_olc – Contains the following scripts:
     - OLC_550_to_600 – Upgrades Online Charging tables
     - olc1_processed_trx_tmp_cre.ttsql – Creates a temporary table in the new structure
     - generate_olc1_processed_trx_tmp.sql – Transfers data from the old table to the temporary table
     - generate_olc1_processed_trx.sql – Transfers data from the temporary table to the new table
   - upg_rt – Contains the following scripts:
     - USG_500_to_550 – Upgrades Rating usage tables
     - CTS_500_to_550 – Upgrades Rating customer tables
     - generate_rated_event.sql – Transfers data from the temporary table to the new table
     - generate_rated_event_tmp.sql – Transfers data from the old table to the temporary table
     - generate_rejected_event.sql – Transfers data from the temporary table to the new table
     - generate_rejected_event_tmp.sql – Transfers data from the old table to the temporary table
     - rated_event_tmp_cre.ttsql – Creates a temporary table in the new structure
     - rejected_event_tmp_cre.ttsql – Creates a temporary table in the new structure
   - Dmp – Used for dump files
   - Log – Used for log files
5. If the upgrade scripts need to be modified according to the account's CC state (versions 5.5 and 6.0), edit the scripts as described above.

Adjusting Tiger Profiles for Version 6.0

The following subsections provide instructions for adjusting Tiger's environment profiles for version 6.0.

Defining a Simplified Environment Profile

Version 6.0 supports a simplified database structure. This means that all areas (app, ref, opr and sec) are located in a single database schema.

Two scripts have been supplied to implement this change. These scripts insert data into the following Tiger repository tables:

- DBCONFIG_SKL
- DBRELATION_SKL
- ENV_PROFILES_SKL
- TIGER_PROFILES_TYPES

The insert automatically defines a new environment profile for Tiger, called test_global_connect. This profile contains:

- One owner containing all database objects
- One connect account containing synonyms and full permissions (select, insert, update, and delete) on the owner objects

To move from the old database structure to the new structure:

From the command line:

1. Modify the scripts to adjust them to your account's naming convention.
2. Change the directory (run cd) to:
   $install_dir/Scripts/
3. Using SQLPLUS, connect to the Tiger repository user account:
   <tiger_username>/<tiger_password>@<db_instance>
4. Run the following script: @test_global_connect.sql – To define the simplified environment profile with one owner and one connect.

Adding New Streams (Areas) and Profiles to the Tiger Repository

In version 6.0, four new streams (areas) have been added:

- WL – Wireline
- NA – North America
- APPLCL – For Audit & Control objects that must be created in every database instance
- APPENG – For Audit & Control objects that must be created in one database instance

The new streams need to be added to existing profiles, since the simplified environment profile already includes the new stream (area) definitions.

To add new stream definitions to the CCOBJAREA table:

From the command line:

1. Change the directory (run cd) to:
   $install_dir/Scripts/
2. Using SQLPLUS, connect to the Tiger repository user account:
   <tiger_username>/<tiger_password>@<db_instance>
3. Run @Add_Abp_New_Areas.sql to define the new areas.

To add new stream definitions to the existing Tiger profiles:

From the command line:

1. Change the directory (run cd) to:
   $install_dir/Scripts
2. Using SQLPLUS, connect to the Tiger repository user account:
   <tiger_username>/<tiger_password>@<db_instance>
3. Run the following script:
   - @Add_for_Global_Dev sql – To add the new area definitions to the Global development profile
   - @Add_for_Prod.sql – To add the new area definitions to the production profile

   This script inserts data into the following Tiger repository tables:
   - DBCONFIG_SKL
   - DBRELATION_SKL
   - ENV_PROFILES_SKL

To add new profiles to Tiger repository:

- Run
  - @test_part_dist sql – To create new Cycle Division profile
  - @ test_global_conn_sel.sql – To create new Master profile
  - @prod_for_test.sql – To create new production like profile (must be run after Add_for_Prod.sql)

Preparing DB Schema Upgrade

Dependency: Configure and upgrade CC

Tiger uses the pack/unpack command line utility to put the required data into Tiger repository tables. After using this utility, Tiger repository will include all the modules of the packed version, and the most recent cc gdd definitions of the core layer.

This action is optional in CDE environment. It can be used when there is a need to build DB environment with version 600 and the CC is not ready yet.

Unpacking Version 600 (Core Only)

1. Change directory to the Tiger home directory.
2. To unpack, perform the following steps:
   a. Create a temp directory under the Tiger account:
      ```
      mkdir temp
      ```
   b. Run PackTigerDbRep as follows:
      For CDE the mode used in the pack is INITIAL which populates the Tiger repository to reflect copy of ABP 600 core as Initial patch 0.
      ```
      PackTigerDbRep -tigerdb
      <tiger_username>/<tiger_password>@<db_instance> -
      tempdir <the created temp dir> -unpack
      $install_dir/PackRepository/<name of supplied
      packing file>
      ```
   c. Be sure to use full path for the –unpack parameter.
      For example:
      ```
      PackTigerDbRep -tigerdb tiger_dls/tiger_dls@stss9i
      -tempdir /tgruser.v894/dls/tgr/dlstgr/Temp/ -unpack
      $install_dir/PackRepository/PACK_TIGER_600_ABSS9IU_
      ABP600_PATCH52_INITIAL_201004_114702.tar.gz
      ```

Configuring Tiger for CDE

The unpack loads into the Tiger repository the CC home definition of the core layer.

To update the CDE environment definition with the relevant CC home, perform the following steps:

1. Connect to the Tiger repository user, e.g. <tiger_username>/<tiger_password>@<db_instance> using SQLPLUS.
2. Update CCINFO set CC_HOME_DIR='<Full path of the CC home dir>' where CC_VERSION='600';
3. Update CCMODULES set CC_HOME_DIR='<Full path of the CC home dir>' where CC_VERSION='600';

To find the full path of the CC home dir type: echo $CCPROJECTHOME

Preparing DB Patch

This part is necessary only if you have customized GDD objects in the CC proj area after a successful build. If there is no customization part in the proj area, you can skip the database patch.

Loading GDD Files into Tiger

After updating the CC with the customization changes there is a need to load the updates into the Tiger repository.

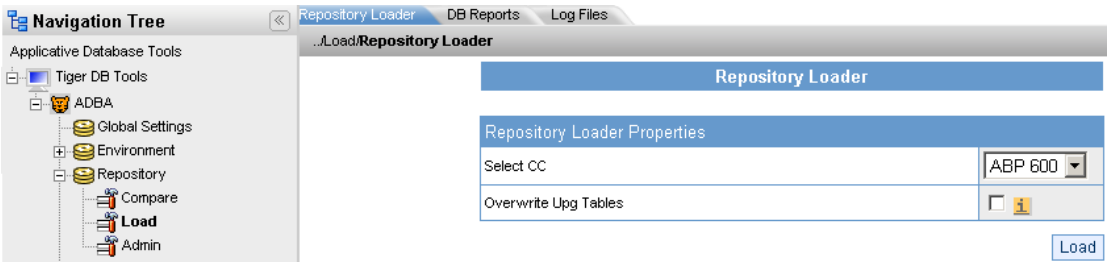
To load the GDD objects into Tiger repository:

- Connect to the Tiger URL.

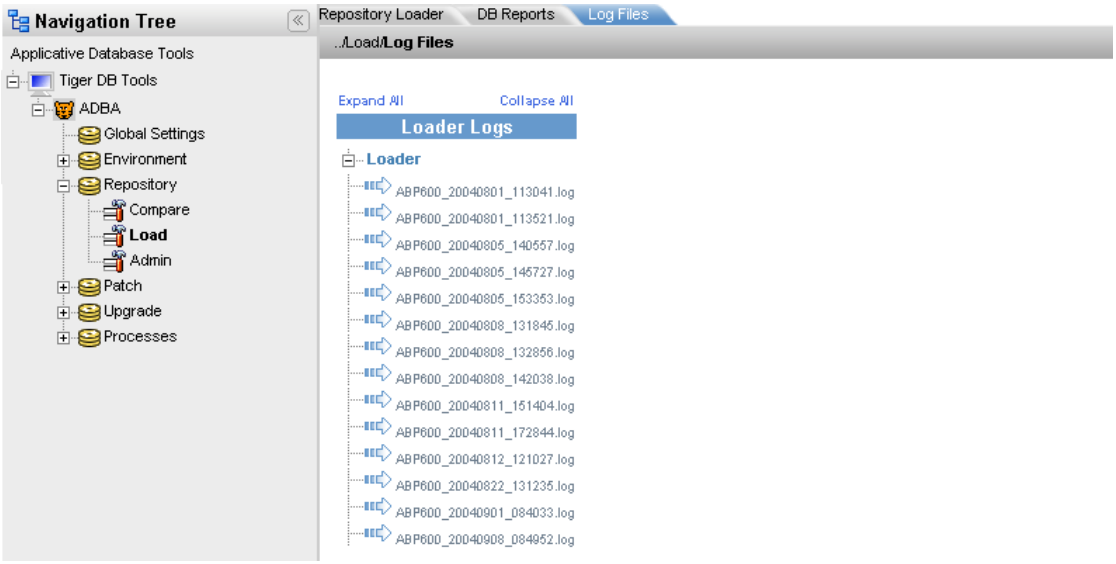
To load the new repository:

From Tiger:

1. Connect to Tiger the URL.
2. Expand the navigation tree from the Repository node to the Load node.
3. Select the Repository Loader tab. The following screen is displayed:



4. From the Select CC dropdown list, select **ABP 600**.
5. Click **Load**.
6. Check for errors in the Load log files as follows:
   a. Select the Log Files tab. The following screen is displayed:



   b. Click the link of the relevant log file. The file is displayed in the workspace.
   c. Alternatively, view the log files in the Tiger UNIX account located in ~/TIGER/Logs/Rep/Loader/*.log

   The following errors can be ignored:

   - *ttsql
   - *_v.sql
   - *_fld.sql

   Any other errors must be examined and handled.

Creating Db Patch #1

Preparing DB patch is required only if the load described above is not patch 0 (meaning patch #1, Load after unpack).

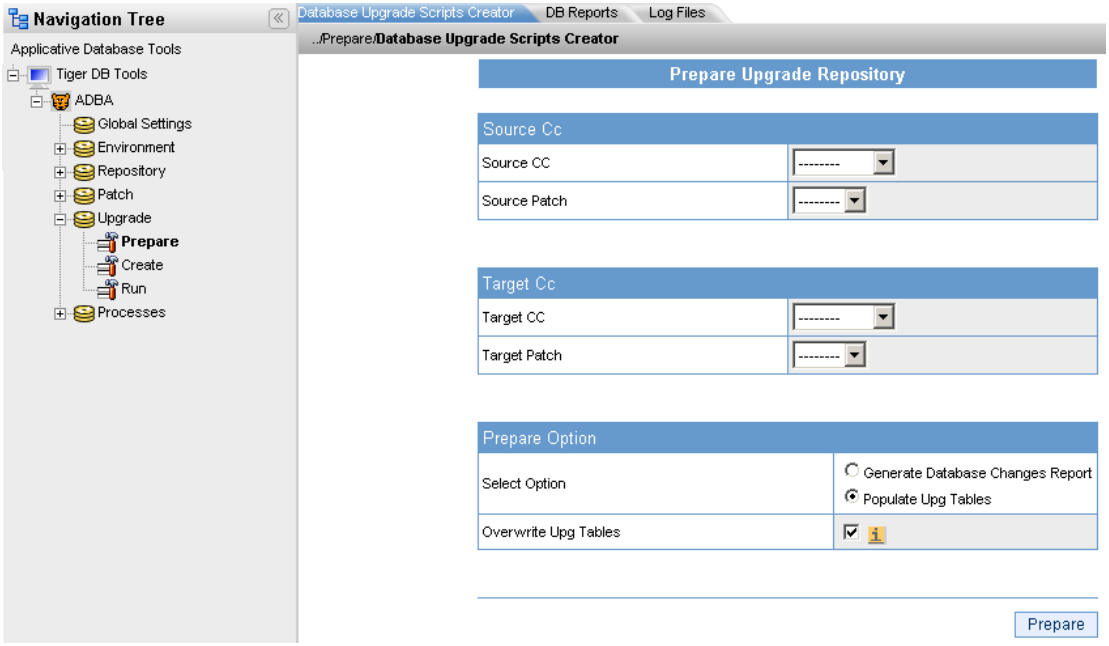
After the load make sure to update the upg tables with the default values and pre/post scripts if needed.

Preparing the Upgrade Kit

The upgrade script preparation process compares the two CC states that were loaded, and collects all the information regarding default values and conversion scripts from patches run during the previous version.

To prepare the upgrade script:

1. Connect to the Tiger URL
2. Expand the navigation tree from the Upgrade brunch.
3. Click on the Prepare brunch.
4. Select the Database Upgrade Scripts Creator tab. The following screen is displayed:



5. In the Source CC area, set the following:
   a. Source CC – Select **ACM550**
   b. Source Patch – Select the required patch
6. In the Target CC area, set the following:
   a. Target CC – Select **ABP600**
   b. Target Patch – Select the required patch

7. In the Prepare Options area, set the following:
   a. Select Option – Select **Populate Upg Tables**
   b. Select Overwrite Upg Tables
   c. Click **Prepare**.
8. To view the database changes, select the DB Reports tab and expand the relevant directory. For example, PREPARE_ABP600_20040815_154037, where the suffix is the time stamp.

Preparing the Prescripts and Postscripts

To run the prescripts and postscripts as part of the upgrade and Sync Data processes:

From the command line:

1. Copy the upgrade prescripts and postscripts from the Delivery Contents to the Tiger UNIX account, under:
   Upgrade/Tiger_550_600_upg/Pre_Post_550_600
2. Edit the prescript and postscript with the correct PATH_TO_FILES – Set the export PATH_TO_FILES with the relevant machine details.

From Tiger:

1. Expand the navigation tree from the Processes node to the Pre/Post Scripts node.
2. In the Processes list, click **Upgrade Runner**.
3. In the Pre/Post Scripts area, set the following:
   - Run Mode – From the drop-down list, select **Pre** to run a prescript.
   - Script Name Full Path – Enter the required script:
     Prescript – <full_path>/ Pre_Post_550_600/_
     Pre_Script_550_600.ksh
4. Click **Add**.
5. Click **Save Changes**.
   The postscript must run after SyncData, and not as a post-upgrade script.

   For further details regarding the SyncData option, refer to "Updating Resource Management.
   This activity must be performed before you can upgrade the data. Therefore, instructions for this activity are given at the end of this Chapter.
6. Expand the navigation tree from the Processes node to the Pre/Post Scripts node.
7. In the Processes list, click **Sync Data**.

8. In the Pre/Post Scripts area, set the following:
   - Run Mode – From the drop-down list, select **Post** to run the postscript.
   - Script Name Full Path – Enter the required script:
   - Postscript – <full_path>/ Pre_Post_550_600/_ Post_Script_550_600.ksh
9. Click **Add**.
10. Click **Save Changes**.

Identifying Renamed Objects and Other Manual Changes

Updating the UPG_OBJECTS Table

To maintain tables that were dropped in version 6.0 for the upgrade process, the action in Tiger's upg_objects table must be changed from DROP to RENAME NO CONST.

Since Tiger does not recognize such cases, the ADBA must carefully review the DB Report to identify these changes.

The Pre_Update_UPG_OBJECTS.sql script (located under $install_dir/Upgrade/Tiger_550_600_upg/Pre_Post_550_600) handles these actions for core objects by updating the upg_objects table with the RENAME NO CONST action, old_object_name and new object_name.The script also manipuldate the data for other tables that required manual changes. Please see the scripts for specific reason for each table.

Any customized objects that were dropped but are required for the upgrade must be added to the script for the final upgrade.

The *RENAME NO CONST* action drops all table constraints, indexes and renames the table.

During the upgrade process, all renamed tables receive the _UPG_55_60 suffix. For example, TABLE1 is renamed to TABLE1_UPG_55_60.

To run the script and update the Tiger repository:

1. Change the directory (run cd) to the $install_dir/Upgrade/Tiger_550_600_upg/Pre_Post_550_600
2. Using SQLPLUS, connect to the Tiger repository user account:

   <tiger_username>/<tiger_password>@<db_instance>
3. Run the @Pre_Update_UPG_OBJECTS.sql

Updating the UPG_COLUMNS Table

Columns that were renamed between version 5.5 and 6.0, before Tiger and Fox supported the Rename Column option, must be handled manually in the upg_columns table. The Pre_Update_UPG_COLUMN.sql script modifies the entries in upg_columns for such cases.

Refer to the script to view the list of columns that were renamed, and the related tables.

To run the script and update the Tiger repository:

1. Change the directory to:
   $install_dir/Upgrade/Tiger_550_600_upg/Pre_Post_550_600
2. Using SQLPLUS, connect to the Tiger repository user account:
   <tiger_username>/<tiger_password>@<db_instance>
3. Run the @Pre_Update_UPG_COLUMNS.sql script.

Handling Dynamic Partitions in Tiger Repository for Version 550

In version 550 dynamic partitions were not supported by Tiger.

To avoid rebuild of dynamically partitioned tables, the partition details should be updated in the Tiger repository.

To update the Tiger repository with details for dynamic partitions:

1. Change the directory to:
   $install_dir/Upgrade/Tiger_550_600_upg/Pre_Post_550_600
2. Using SQLPLUS, connect to the Tiger repository user account:
   <tiger_username>/<tiger_password>@<db_instance>
3. Run @PreUpdatePartitionDetails.sql which creates the package `update_partition_details_pg` and tables `part_vals_55_60 , part_vals1_55_60`
4. From sqlplus run the package update_partition_details_pg.upd_part_dets('U') to update the Tiger repository as follows:

   ```
   exec update_partition_details_pg.upd_part_dets('U')
   ```

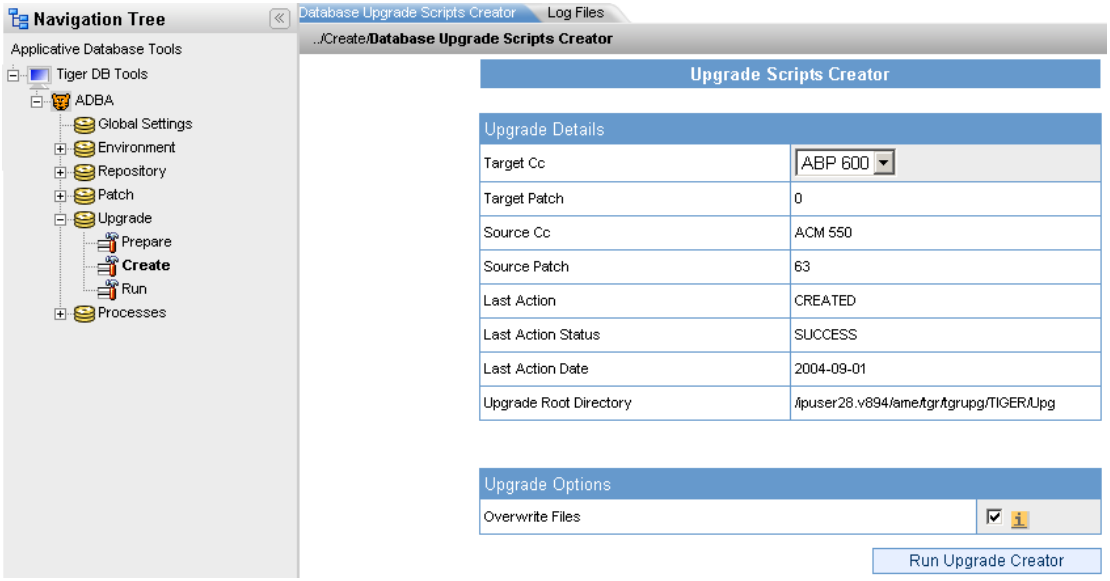
5. The package updates this tables:
   - CCOBJECTS
   - PARTITION_VALUES

Running Upgrade Creator

Run Upgrade Creator after all information required for the upgrade is updated in the UPG tables.

To run Upgrade Creator:

1. Connect to the Tiger URL
2. Expand the navigation tree from the Upgrade node to the Create node.
3. Select the Database Upgrade Scripts Creator tab. The following screen is displayed:



4. From the Target CC dropdown list, select **ABP 600**.
5. Click **Run Upgrade Creator**.

Running the Upgrade

For running the upgrade on the environment, refer to Amdocs Billing Platform (Runtime).

Post-upgrade Activities

Following the completion of the upgrade process, some post-upgrade activities are performed, such as removing version 5.5 3rd party and sanity checks on the CDE environments.

The following diagram illustrates these post-upgrade activities.

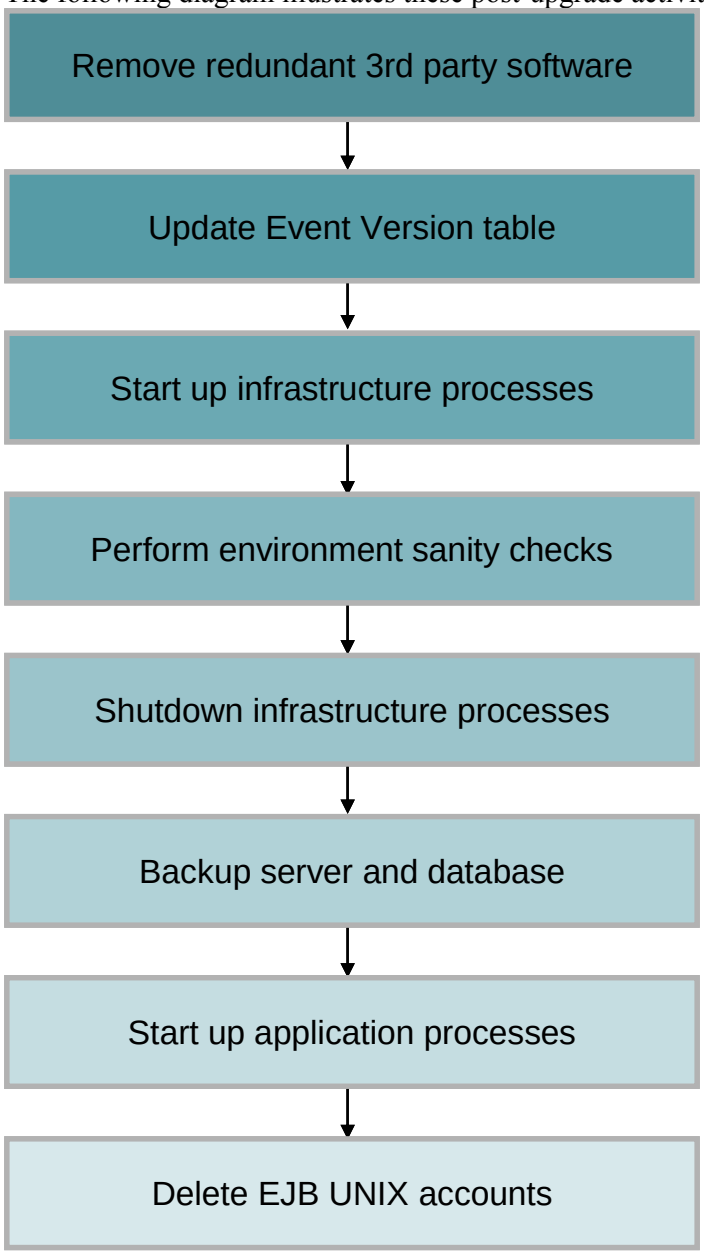


Figure 5-1: Post-upgrade Activities Workflow

Remove Redundant Third-party Software

Remove any version 5.5 third-party software that is no longer used.

Appendix A. Upgrading UAMS

This section details the changes that are specific to upgrading UAMS in the ABP.

Please read the UAMS 0.50 Upgrade Guide to learn about the other, generic, changes that are also relevant to ABP.

New and Updated Files

1. Updated uams.jar
2. Updated UamsSecurityProviders.jar
3. Updated Sonar templates
4. New UamsForms.war

Configuration Parameters

Use the following values for the -Damdocs.uams.config.resource property.

1. For Security Server: res/acm/secsrv (it includes the IP configuration)
2. For Application Server (BE): res/acm/appsrv
3. For Web Server (FE): res/acm/websrv (it includes the SP configuration)
4. For Full (secsrv+web+app together): res/acm/full
5. For No security configuration : res/acm/websrvNosec

BOH Support

In this version UAMS added a new Administration Delegation feature. This feature uses the BOH tree.

In order to use the BOH APIs one should follow theses steps after completing the previous security server installation guides:

Note that all files are located in the UAMS version tools/boh directory or in the ABP cc.

1. Add the ***ggn_classes.jar*** to your classpath
2. Make sure you have the ***jutil.jar*** in your classpath
3. Add the following files to your amdocs.system.home directory:
   - ACMArchError_en.xml
   - GN1Config.properties
   - MessageHandlingServices.dtd
   - messageHandlingServices.xml
4. Add the ROOT_NS_NAME=0 property to your local properties file and set the id of the root BOH.

Database

1. Two new tables were added: SEC1_RESOURCE and SEC1_POLICY to support the new authorization and administration delegation features.

   The SEC1_LOG table was changed to support ABP requirements (add app_id to the event report) and UAMS new events. It is better to save the old table under a different name and install it again using the new sqls.

   Please see Installing and Populating UAMS Database in the UAMS Upgrade Guide.

2. Please note that due to the new feature that supports the APB BOH tree, the root namespace name that was used in UAMS 0.49 ($) should be changed to the new root namespace name that is equal to the BOH root id.

   In order to do that, open the SEC1_NAME table and find the raw in which the NAME column value = $. Change (manually) the value of the NAME and PATH columns to the id of the root BOH node (usually it is 0).

   After this change the database can no longer work with the UAMS version 0.49.

UAMS SSO Filter Configuration

web.xml Configuration

The web.xml should be changed to use the new filter implementation and to use a enw. The application's web.xml should contain the following lines:

```
<filter>
<filter-name>UamsSSOFilter</filter-name>
<filter-class>

amdocs.uamsimpl.client.login.sso.servlet.sp.UamsSSOFilter
</filter-class>
      </filter>
<filter-mapping>
<filter-name>UamsSSOFilter</filter-name>
<url-pattern>/*</url-pattern>
      </filter-mapping>
<servlet>
<servlet-name>UamsLoginCompleteServlet</servlet-name>
  <servlet-class>
amdocs.uamsimpl.client.login.sso.servlet.sp.UamsLoginComp
leteServletImpl
        </servlet-class>
      </servlet>
```

```
<servlet-mapping>
<servlet-name>UamsLoginCompleteServlet</servlet-name>
<url-pattern>/UamsLoginCompleteServlet</url-pattern>
</servlet-mapping>
```

UAMS Properties

By default, these params should not be configured, they will automatically get the values of the SEC_SRV_CONN parameter.

- IDNTT_PRVDR_HOST=<ip_host_name>
- IDNTT_PRVDR_PORT=<ip_port_name>

If the IP will be separated from the security server, these parameters will have to be set.

You may disable the filter by adding the SERVLT_LOGIN_DISABLE=1 property to your local uams.properties file.

(0 = false, 1 = true)

Upgrade Steps

1. Refresh the storage to have updated UAMS files
2. Drop the SEC1_LOG table (or save it under a different name) and run Fox to install the two new tables and update the SEC1_LOG table.

Update the environment with new Sonar templates to get an updated config.xml file and updated uams.properties (use Sonar 6.0).

Appendix B. DVM, RDM Persistence Tools - Migration Script

Motivation

Reference tables datalayers and database views used to have stand-alone GUIs for editing their mappings and methods (Database View Manager - DVM tool and Reference Data Manager – RDM tool).

In version 6.0 a dStudio plug-in has been developed for editing database views (DVM DBViews). Existing DBViews should be upgraded for dStudio support.
The RDM tool is declared obsolete and should no longer be used. Existing datalayers should be converted to DVM format. (DVM will generate RDM style code for RDM converted DBViews for backward compatibility).

Accounts which are upgrading from previous versions and have existing database views and/or reference datalayers should run the migration script.

What Does the Script Do?

1. Checkout existing database view and reference datalayer source files.
2. Convert database views to new format and rename to *.dvm (instead of *.xml)
3. Convert reference datalayer to new database views. Rename to RT*.dvm (instead of DL*.xml)
4. Create a default rdm.properties for usage with dStudio.
5. Check-In the new files (under a specific Xtra-C package)
6. Remove Item for old database views and reference datalayer XML files. (Deleted versions will be added to the Xtra-C task)

Migration Steps

1. Create an Xtra-C task
2. Prepare conversion.properties. See details below.
3. Execute the script. See script usage.
4. Validate migration products.
5. If migration was validated, promote the Xtra-C task

Migration Validation

After running the migration process, check the Xtra-C task for the following versions:

1. New version for each original database view source (with *.dvm extension) – in the viewSource topic
2. New version for each original reference datalayer source (with RT prefix and *.dvm extension) – in the viewSource topic
3. Deleted version for each original database view source
4. Deleted version for each original reference datalayer source
5. New version of rdm.properties – in the building-block root topic

Before running the migration script a request should be applied to Xtra-C implementation team for applying the latest command line fixes on the specific machine.

Script Usage

dvm_conversion_cust.tcsh < Migration property file > < XtraC user > < XtraC password > < XtraC broker > < XtraC Project > < XtraC Task name >

Example:

dvm_conversion_Cust.tcsh conversion.properties scott unix11 ep67001a/upg Tasks 600 TS - CM - CR102 - IL - DVM Migration

Migration Property File

The following properties should be included in the migration property file referred in the command line:

- amdocs.persistence.conversion.dvmProperties
  Full path to dvm properties file (also known as “master file”) See content below.
- amdocs.persistence.conversion.viewPackage
  Java package for generated database view classes
- amdocs.persistence.conversion.viewDataPackage
  Java package for generated database view data classes
- amdocs.persistence.conversion.localViewSource
  Full path to local topic of database view source files.
- amdocs.persistence.conversion.viewsourcePackage
  Java package of existing view source files.
- amdocs.persistence.conversion.rdmProject
  The full path to the previous version RDM project xml file. The project file should reference a valid master file (rdm properties)
- amdocs.persistence.conversion.rdmProperties
  Full path to rdm properties file. See content below.

- amdocs.persistence.conversion.rdmsourcePackage
  Java package of existing datalayer source files.

Example:

amdocs.persistence.conversion.dvmProperties=/amhuser23.p803a/ip/cm/scott/DVMConversion/dvm.properties

amdocs.persistence.conversion.viewPackage=amdocs.jfd.db.views

amdocs.persistence.conversion.viewDataPackage=amdocs.jfd.db.views.data

amdocs.persistence.conversion.viewsourcePackage=amdocs.jfd.db.viewsource

amdocs.persistence.conversion.localViewSource=/amhuser23.p803a/ip/cm/scott/bb/gjfde/v60_0/amdocs/jfd/db/viewsource

amdocs.persistence.conversion.rdmProject=/amhuser23.p803a/ip/cm/scott/DVMConversion/ReferenceData600.xml

amdocs.persistence.conversion.rdmProperties=/amhuser23.p803a/ip/cm/scott/DVMConversion/rdm.properties

amdocs.persistence.conversion.rdmsourcePackage=amdocs.jfd.refdata.mapping

DVM Properties

The following entries are required for running the migration script

- amdocs.persistence.factoryName
  Customization layer DVM factory class name
- amdocs.persistence.factoryPackage
  Customization layer DVM factory class package
- amdocs.persistence.controlFields
  List of control fields names
- amdocs.persistence.datatypeDefinitionPaths
  Full path to the customization project “cnt” folder

Example:

amdocs.persistence.factoryName=CM9ViewFactory

amdocs.persistence.factoryPackage=amdocs.csm3g.db.views

amdocs.persistence.controlFields=SYS_CREATION_DATE;SYS_UPDATE_DATE;OPERATOR_ID;APPLICATION_ID;DL_SERVICE_CODE;DL_UPDATE_STAMP

amdocs.persistence.datatypeDefinitionPaths=/upguser/upg/tst/scott/proj/ccm600V64/cnt

RDM Properties

The following entries are required for running the migration script

- amdocs.persistence.factoryName
  Customization layer RDM factory class name
- amdocs.persistence.factoryPackage
  amdocs.csm3g.refdata.rdm

The following entries are optional and should be included only if they were included in the previous version master file. (If existed, the values should be copied from the previous master file.)

- amdocs.persistence.defaultMarket
- amdocs.persistence.preparedStatement
- amdocs.persistence.connectionPool
- amdocs.persistence.defaultLocale
- amdocs.persistence.commit

Example:

amdocs.persistence.factoryName=CM9ReferenceDatalayerFactory

amdocs.persistence.factoryPackage=amdocs.csm3g.refdata.rdm

amdocs.persistence.connectionPool=DataSourceGetter