COMVERSE

Comverse® ONE™

3.5 TR1.0

Network Interfaces and Notifications Guide

ComONE-3.5-TR1.0-NING-2011-04-29

Notice

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

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

Revision History

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

Date	Chapter	Description
04/29/2011		Initial publication.

Contents

Figures

Tables

Notational Conventions

NOTE Useful information appears in this format.

SEE Provides direction to important information

IMPORTANT Important information appears in this format.

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

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

Table 1 Notational Conventions

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

Special Markers

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

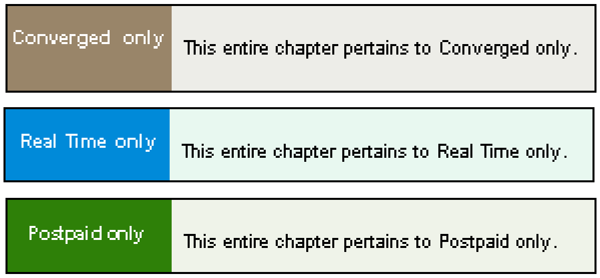
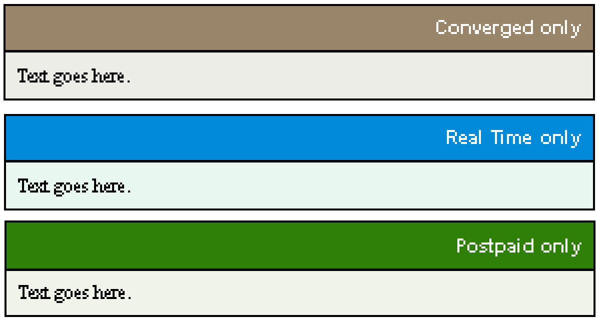
Table 2 Labels in Markers

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

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

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

Table 3 Types of Markers

Marker	Example	Description
Alert	 Converged only This entire chapter pertains to Converged only. Real Time only This entire chapter pertains to Real Time only. Postpaid only This entire chapter pertains to Postpaid only.	- Placed at the beginning of an entire chapter that pertains only to a specific derivative. - Placed just before a table that partially or entirely pertains only to a specific derivative.
Block	 Converged only Text goes here. Real Time only Text goes here. Postpaid only Text goes here.	A shaded box that encloses sections of documentation that pertain only to a specific derivative.
Flag	Converged only Real Time only Postpaid only	- Designates a shaded table row whose contents pertain only to a specific derivative. - In a bulleted list, designates an item that pertains only to a specific derivative.

Comverse ONE Documentation List

NOTE this is a comprehensive list. As such, it may include documentation for products which you have not licensed.

The documents described below reference the Comverse ONE solution products. All documentation available with the Comverse ONE solution is described in the following pages, organized by the following categories:

- Infrastructure Domain
- Rating, Charging, and Promotions Domain
- Billing and Financials Domain (Converged only)
- Customer and Order Management Domain (Converged only)
  - Customer Relationship Management
    (Sale Force Automation, Case Management, Campaign Management)
- Mediation and Roaming Solutions Domain
- Self-Service Solutions Domain

NOTE Read the relevant Solution Description first to get an overview of your Comverse ONE solution. It gives an overview of the functionality in each product domain and also includes cross-references to the user documentation that provides more detailed information about the functionality.
There are two such documents and they are listed under the Infrastructure Domain heading below.

- *Converged Billing & Active Customer Management Solution Description*
- *Real-Time Billing & Active Customer Management Solution Description*

Infrastructure Domain

Download every document in the Infrastructure domain if you purchase the Comverse ONE solution. Documentation for this domain includes the following (in alphabetical order):

- *Alarms Reference*
  Contains tables of alarm IDs, descriptions, likely causes, and recommended resolutions for systems and components.

- *Back Office Administration GUI Guide*
  Provides information about the BackOffice subsystems for Inventory Administration, Address Management and Bulk Operations.
- *Converged Billing & Active Customer Management Solution Description*
  General overview of the Comverse ONE Converged Offer and the functionality available in each domain.
- *Database Reference*
  Describes all database tables and fields in detail.
- *Disaster Recovery Operations Guide* (Optional Module)
  The Disaster Recovery Operations Guide serves as both a technical overview of the optional Disaster Recovery solution and as a guide which details the operational procedures for failover, switchover and switchback provided by the solution.
- *Glossary*
  Provides a list of terms used specifically for the Comverse ONE solution
- *Investigation Units and Financial GUIs Guide*
  Describes the GUI-based tools used for investigating and troubleshooting various financials related processes: payments, bill invoices, refunds, and incomplete data work entries
- *Operation Reference*
  Describes the processes in the Comverse ONE solution.
- *Platform Operations Guide*
  Describes the back-end operations and maintenance functionality of the core Comverse ONE solution components. Includes AIX/HACMP platform and cluster operations, Linux/Veritas platform and cluster operations, backup/recovery, shared storage and fiber switch operations, and tape backup operations.
- *Product Catalog Overview*
  Provides a high-level description of the Comverse ONE solution Product Catalog, which is the primary mechanism for creating, configuring, managing, and propagating Product Catalog versions.
- *Product Catalog User Guide*
  Instructions on using the Product Catalog application to define and manage all aspects of Service provisioning.
- *Real-Time Billing & Active Customer Management Description*
  General overview of the Comverse ONE Real-Time Offer and the functionality available in each domain.
- *Schedulable Entity Reference Manual*
  Documents all the jobs, monitors, and workflows, for each component.
- *Security Platform Operations Guide*
  Technical overview of the security platform and information on how to provision and administer the platform.
- *Security Server API Guide*
  Provides an overview of the interfaces exposed by the Java-based Security SDK API, which client applications can leverage to access various security services, such as authentication, authorization, auditing, key management, and credentials management. Also provides information on the Security Web Services API, which provides interfaces to a subset of Security Server commands (Identity Management commands).
- *Signaling Gateway Unit Guide*
  Describes the hardware, installation, configuration, and maintenance of the Signaling Gateway Unit (SGU) used to connect Comverse real-time systems to the SS7 signaling network using either traditional SS7 protocols or Sigtran (SS7 over IP).
- *System Measurements Guide*
  The Comverse ONE Solution automatically collects statistical data from the Service Logic Unit (SLU) and the Service Gateway Unit (SGU). This includes service statistics on the SLF layer and

platform data on the IPF layer.
This guide describes the format and location of this measurement information and provides a description of the meaning of the data.The measurement data can be used to create reports. It can also be imported into other applications (such as Excel) to be viewed.

- *Unified API Guide*
  General overview of the Unified API, a brief description of its architecture, and information about:
  - Framework classes and the functionality they provide
  - Two standard interfaces provided with the Unified API (client SDK and web services)
  - A subset of Unified API business methods most commonly used
- *Unified Platform Guide*
  Technical overview of the Unified Platform and information on the procedures to manage core systems operations in the Comverse ONE solution.

Rating, Charging, and Promotions Domain

Documentation for this domain includes the following (in alphabetical order):

- *Batch Provisioning Utility Guide*
  - The *CC Batch* utility enables bulk creation of recharge vouchers and subscribers.
  - The *Bulk Provisioning* Utility enables bulk creation of anonymous accounts to support the pre-activation of pre-paid SIM cards.
- *Call Flows Reference*
  Call flows detail the logic flow of specific scenarios. Multiple access numbers can map to the same call flow. Different resellers have the option to publish different numbers but share the same logic.
- *Charging Interfaces Guide*
  Describes the four interfaces that enable external services to support real-time authorization, rating, and charging for transactional usage: (1) the Event Charging Interface, a simple TCP/IP-based interface, (2) Open Services Access (OSA), (3) a Diameter-based interface version enhanced to take advantage of features of the Comverse ONE solution, and (4) a Diameter-based interface packet-switched version.
- *Customer Care Client Provisioning Guide* — Real-Time
  Detailed task-oriented instructions for using Customer Care Client.
- *Diameter Gateway Unit Guide*
  Describes the hardware, installation, configuration and maintenance of the Diameter Gateway Unit (DGU) used to connect Comverse real-time systems to external services, using the diameter protocol over IP.
- *Network Interfaces and Notifications Guide*
  Describes the operation, features, and provisioning of notifications, CAMEL-enabled services, and USSD-enabled services.
- *Network Self-Care Guide*
  Describes the configuration, structure, and features.
- *Rating Technical Reference*
  Describes the Unified Rating Engine, which is the subsystem responsible for gathering incoming CDRs and processing them for billing.
- *Reports and Data Extracts Guide* — Real-Time
  Describes the real-time Operational Reports Interface (ORI) and the Data Warehouse Extract Utility.
- *Recurring–Non-Recurring Charges Server Guide*
  Describes all processes commonly available through the Recurring —Non-Recurring Charges Server.
- *Voucher and Recharge Guide*
  Describes the process by which subscribers add funds to accounts using recharge vouchers through IVR, interaction with Customer Service, and other methods. Provides details of the Recharge Control

Table, which allows resellers to provision the effects of recharges so that bonuses, discounts, and other changes to offers can result from a successful recharge. Also describes the Card Generator software used to create batches of vouchers and calling cards.

Billing and Financials Domain (*Converged* only)

Documentation for this domain includes the following (in alphabetical order):

- *Advanced Statement Numbering Guide*
  Describes how to configure and use Advanced Statement Numbering.
- *Billing Reports and File Layouts User Guide*
  Describes control reports and other file formats.
- *Billing Technical Reference*
  High-level descriptions of billing architecture, administration, bill generation and formatting, and system parameters
- *Collections Guide*
  Contains information on configuring Collections database tables, running the Collections module, and using the Collections interface.
- *Configurator Guide*
  Describes how to install and use the Configurator.
- *Invoice Designer Strings and Filters Reference*
  Describes the static strings, dynamic strings, and filters in the Invoice Designer.
- *Invoice Designer Technical Reference*
  Describes how to configure and run Invoice Designer.
- *Invoice Designer User Guide*
  Describes the Invoice Designer and how to perform the tasks needed to create an invoice template.
- *Journals Guide*
  Describes the theory, configuration, and running of Journals processes.
- *Miscellaneous Configurable Entities*
  Instructions for configuring late fees, adjustments, and several other database entities used in postpaid and converged billing.
- *Process Workflow Orchestration Guide*
  Describes the command-line entries and the default queries for running billing-related processes via the Unified Platform.
- *Taxation Guide*
  Describes the configuration, operation, structure, and features of Taxation.

Customer and Order Management Domain (*Converged* only)

Documentation for this domain includes the following (in alphabetical order):

- *Application Integrator Adapter Developer Kit User Guide*
  Provides information necessary for the development of custom Application Integrator adapters.
- *Application Integrator Add/Copy Header User Guide*
  Describes the adapter that adds or copies header information in messages.
- *Application Integrator Aggregator Adapter User Guide*
  Describes the adapter that aggregates multiple input messages as a single composite output message.
- *Application Integrator File Adapter User Guide*
  Describes the configuration process and rules for the file adapter.
- *Application Integrator CORBA Adapter (JacORB) User Guide*
  Describes the elements and uses of the Application Integrator client and server Common Object Request Broker Architecture (CORBA) adapters for JacORB.
- *Application Integrator Filemover Adapter User Guide*
  Describes the use and configuration of the adapter, which is used to copy or move files from one machine to another.
- *Application Integrator Generic Services User Guide*
  Describes the Null adapter, Trash adapter, and Initiator adapter generic services.
- *Application Integrator HTTP Adapter User Guide*
  Describes the use and configuration of the adapter which provides an interface between HTTP clients and the ApplicationIntegrator.
- *Application Integrator IPDR Adapter User Guide*
  Describes use and configuration of the I adapter which converts the “compact encoding” form of IPDR billing record documents into a form easily parsed by the ApplicationIntegrator message broker.
- *Application Integrator JMS Adapter User Guide*
  Describes the use and configuration of the adapter, which is used with edge systems that transmit or receive JMS messages.
- *Application Integrator KSI Adapter User Guide*
  Describes the use and configuration of the adapter, which is used with edge systems that transmit or receive data formatted according to the Kenan Standard Interface (KSI) protocol.
- *Application Integrator Operator Guide*
  Describes the commands that operate the Application Integrator at creation and runtime.
- *Application Integrator Python Adapter User Guide*
  Describes the use and configuration of the adapter, which enables a user to run a *Python* script from within an integration.
- *Application Integrator Retry Adapter User Guide*
  Describes the use of the a dapter to resend messages in case of failed transmissions.
- *Application Integrator SAS Adapter User Guide*
  Describes the use and configuration of the adapter, which is used with edge systems that transmit or receive data formatted according to the *Comptel* Mediation Device Solutions/Subscriber Administration System (MDS/SAS) protocol.
- *Application Integrator Sequence Adapter User Guide*
  Describes the use of the adapter to generate unique sequence numbers for messages.
- *Application Integrator System Administrator Guide*
  Outlines installation, sizing, operation, and administration of the Application Integrator and logging. Describes configuration of the user environment and commands for creation and operation of the Application Integrator.

- *Application Integrator Unified API Client Adapter User Guide*
  Describes the adapter which is used for interfaces based on the Unified API Client.
- *Application Integrator Unified API Server Adapter User Guide*
  Describes the adapter ehich is used for interfaces based on the Unified API Server.
- *Application Integrator URL Client Adapter User Guide*
  Describes the use and confoguration of the adapter which makes it possible for a client to gain access to many kinds of network-accessible resources that are identified by a URL.
- *Application Integrator User Guide*
  Describes creating integration specifications, creating instances of the Application Integrator, and commands for operation of the Application Integrator. Provides a complete user guide for the iMaker compiler.
- *Application Integrator XSLT User Guide*
  Describes the use and configuration of the adapter which is usedwith applications (sometimes called edge systems) that transmit or receive XML- formatted data.
- *Customer Center User Guide*
  Detailed task-oriented instructions for using Customer Center.
- *Inventory Guide*
  Describes the configuration, operation, structure, and features of Inventory.
- *Inventory Replenishment Guide*
  Describes the operation, structure, and features of Inventory Replenishment.
- *Orders Services Guide*
  Describes the structure and features of Orders Services.
- *Request Handling and Tracking and Service Fulfillment User Guide*
  Describes the configuration, operation, structure and features of Request Handling and Tracking and Service Fulfillment.
- *Workflow Developers Guide*
  Helps new users understand the rules-based business process management system so users can create solutions and integrate Workpoint within those solutions.
- *Workflow User Guide*
  Describes the configuration, operation, structure, and features of Workpoint.

Customer Relationship Management

- *Billing Reports and File Layouts User Guide*
  Describes control reports and other file formats.
- *Campaign Management Data Mapping Reference*
  Describes how the data in DataMart is mapped to information in the Comverse ONE Customer database, the Comverse ONE ODS, and the Comverse ONE Sales and Service database.
- *Campaign Management DataMart Reference*
  Contains in-depth technical information on how to configure and populate the data mart used by all Campaign Management applications.
- *Campaign Management Outbound Marketing Manager Reference*
  Describes how to use the Campaign Management Outbound Marketing Manager features and guides you through the program's basic functionality.
- *Campaign Management Quick Implementation Guide*
  Helps novice users get started with implementing Campaign Management. It contains an overview of the product architecture, information on data mart design and creation, an explanation of how extraction works, and procedures for creating web pages, reports, lists, and campaigns.
- *Campaign Management Topic Implementation Guide*
  Provides information for implementers and professional services personnel who are creating applications that will run on an Campaign Management EpiCenter. Summarizes the Campaign Management functionality, architecture, and administration and contains in-depth technical information for configuring the Campaign Management topics required for Campaign Management and analysis.
- *Campaign Management User Guide*
  Provides you with basic information about the Campaign Management applications.
- *Case Management User and Administration Guide*
  Contains detailed information about GUI screens and form fields that appear in the Case Management application. Also provides information on performing general procedures in the GUI and administrative tasks.
- *Customer Center User Guide*
  Detailed task-oriented instructions for using Customer Center.
- *Sales and Service Admin Console User Guide*
  Provides supervisors, managers, and executives with the information to use the Case Management and Sales Force Automation Admin Console application.
- *Sales and Service Application Reference* Contains technical reference information relevant to implementers involved in implementing and customizing CRM applications at customer sites. This book provides the reference context for the procedural information available in the Implementation Guide.
- *Sales and Service Architecture Reference*
  Provides technical information relevant to individuals involved in implementing the Open Architecture and the applications built on the architecture
- *Sales and Service Data Dictionary Reference*
  Includes a listing and description of the tables and columns used to store CRM operational business data. It also includes a description of the naming conventions for the tables. The target audience includes database administrators, application developers, and implementers.
- *Sales and Service IBR Designer User Guide*
  Describes how to use the IBR Designer to create Intelligent Business Rules, which can be used to implement rule-based behavior within your CRM applications.
- *Sales and Service Implementation Guide*
  Provides procedural information relevant to individuals involved in implementing and customizing the core and the Sales and Service applications built on the core.

- *Sales and Service Integration Guide*
  Provides overview and configuration information for the set of tools used to exchange data with a variety of back-end data sources, including generic SQL sources, Java and EJB-based sources, Web services, and other database types.
- *Sales and Service Workflow Designer*
  Explains how to use Workflow Designer, a web-based graphical tool for defining and editing workflows
- *Sales Force Automation User and Administration Guide*
  Contains detailed information about GUI screens and form fields that appear in the Sales Force Automation application. Also provides information on performing general procedures in the GUI and administrative tasks.

Mediation and Roaming Solutions Domain

Documentation for this domain is subdivided into Mediation/Roaming and Revenue Settlements.

Mediation and Roaming

Mediation and Roaming documentation includes the following (in alphabetical order):

- *Collection API Guide*
  Provides the concepts and functions for the Collection Application Programming Interface (CAPI).
- *Data Manager GUI Reference*
  Contains detailed information about GUI screens and form fields that appear in the Data Manager interface
- *GRID Mapping Language Developer Guide*
  Describes the mediation feature components, semantics, and general syntax of the GRID Mapping Language (GML).
- *Installation Guide for HP*
  Describes how to install and configure the application, components, and some third-party applications associated with the HP platform.
- *Installation Guide for HP Itanium*
  Describes how to install and configure the application, components, and some third-party applications associated with the HP Itanium platform.
- *Installation Guide for HP PA-RISC*
  Describes how to install and configure the application, components, and some third-party applications associated with the HP PA-RISC platform.
- *Installation Guide for IBM*
  Describes how to install and configure the application, components, and some third-party applications associated with the IBM platform.
- *Installation Guide for SUN*
  Describes how to install and configure the application, components, and some third-party applications associated with the SUN platform.
- *Mediation and Roaming User Guide*
  Provides information on how to use the GUI interface, including information on using the Data System Manager application pages.
- *Mediation API Guide*
  Contains reference information on using the Mediation API.
- *Roaming Database Reference*
  Provides reference information on the Roaming database.

- *Roaming Setup Guide*
  Describes how to configure the Roaming Setup application pages. It also provides information on working with TAP, RAP, and CIBER statistics.
- *Scripts Guide*
  Provides information on script files, which contain additional instructions on functions for data collection and transmission.
- *Socket-Based API Guide*
  Explains the building applications using the Socket-Based Record Transmission API. Programmers can use the guide to use the records received from the Data system for their own customized downstream application solutions.
- *System Manager GUI Reference*
  Contains detailed information about GUI screens and form fields that appear in the System Manager interface
- *Variable-Length GRID Guide*
  Provides information on how to configure the control files for variable-length GRID.

Revenue Settlements

Revenue Settlements documentation includes the following (in alphabetical order):

- *Comverse Revenue Settlements Billing System Adapter Guide*
  Describes the configuration, operation, and installation for the Billing System adapter.
- *Comverse Revenue Settlements Data Model Guide*
  Overview of data model entities (such as partners, accounts, revenue sharing, and rate schedules) and how to configure them in the database.
- *Comverse Revenue Settlements Database Reference*
  Detailed descriptions of fields and tables in the database.
- *Comverse Revenue Settlements Technical Reference*
  Instructions for installing and operating Revenue Settlements. Also contains processing descriptions.
- *Comverse Revenue Settlements User Guide*
  Instructions for using the Revenue Settlements GUI.

Self-Service Solutions Domain

The Comverse ONE Self-Service Solutions domain consists of the core products plus the optional separately licensed premium products. The core products consist of the following:

- Self-Service Solutions Platform
- Self-Service Solutions Applications

Self-Service Solutions Platform Documentation

The Self-Service Solutions Platform has a comprehensive set of documentation covering the installation, configuration, and use of our products. The documentation set is divided into the following categories:

- **Manuals**: These manuals cover installing and using the platform.
- **Reference**: These reference documents contain information about APIs, databases, configuration files, and so on. These documents are delivered in HTML.

Self-Service Solutions Platform Manuals

Self-Service Solutions Platform manuals include the following (in alphabetical order):

- *Administration Guide*
  Provides operations and maintenance instructions for Web applications using the Self-Service Solutions Platform.
- *Communications Billing and Usage Reference*
  Provides detailed descriptions of the data models and structure of the Self-Service Solutions Platform Communications Billing and Usage (CBU) database.
- *Connectors Development Guide*
  Provides instructions for developing and customizing Connectors of the Self-Service Solutions Platform.
- *Core Module Development Guide*
  Provides instructions for configuring and developing features of the core module of the Self-Service Solutions Platform.
- *Customer Interaction Datastore Reference*
  Provides detailed descriptions of the data models and the structure of the Self-Service Solutions Platform Customer Interaction Datastore (CID).
- *Database Modules Development Guide*
  Provides instructions for configuring, customizing, and developing features of the database module of the Self-Service Solutions Platform.
- *Platform Installation Guide*
  Provides installation and configuration instructions for the Self-Service Solutions Platform.
- *Platform Services Guide*
  Provides instructions for configuring, customizing, and developing features that use the services provided by the Self-Service Solutions Platform.
- *Processors Development Guide*
  Provides instructions for developing and customizing Processors of the Self-Service Solutions Platform.
- *Reports Development Guide*
  Provides instructions for developing and customizing Reports of the Self-Service Solutions Platform.
- *Self-Service Solutions Overview Guide*
  Provides a high-level architectural and functional description of the Comverse ONE Self-Service Solutions. It also includes a detailed description of the concepts and development process to create and deploy Self-Service Solutions.
- *Web Applications Development Guide*
  Provides instructions for configuring, developing, and deploying Web applications that use the Self-Service Solutions Platform.

Self-Service Solutions Platform Reference

Self-Service Solutions Platform reference documentation includes the following (in alphabetical order):

- *Base Logic Manager Reference*
  Describes usage syntax and configuration files for the Base Logic Manager (BLM) APIs. These APIs are the core services of the Self-Service Solutions Platform.
- *CID2CBU Object Mapping Reference*
  Describes the default mapping of Customer Interaction Datastore (CID) and Communications Billing and Usage (CBU) objects.
- *Communications Billing and Usage Reference*
  Provides detailed descriptions of fields and tables in the Communications Billing and Usage (CBU) database.

- *Customer Interaction Datastore Reference*
  Provides detailed descriptions of fields and tables in the Customer Interaction Datastore (CID).
- *Integration Services Framework API Reference*
  Describes usage syntax of the set of APIs to program connectors and other components of the Intelligent Synchronization Framework (ISF).
- *Integration Services Framework Message Cache Reference*
  Provides detailed descriptions of fields and tables in the Intelligent Synchronization Framework (ISF) Message Cache.
- *Integration Services Framework Script API Reference*
  Describes usage syntax of the Intelligent Synchronization Framework (ISF) script APIs to program the ISF connectors.
- *JavaServer Page Framework for Internet Application API Reference*
  Describes usage syntax for the JavaServer Page Framework for Internet Application (JFN) APIs. These APIs are used to build JSPs using the JFN. This framework provides basic application functions and services as the foundation of user interfaces.
- *Logger Message Reference*
  Provides detailed descriptions of the Self-Service Solutions Platform log messages.
- *QRA API Reference*
  Describes usage syntax for the Query, Reporting, and Analysis (QRA) Engine APIs. These APIs are used to build reports.
- *UTIL API Reference*
  Describes usage syntax for the UTIL package used by different components of the Self-Service Solutions Platform. This package contains a set of utilities including the logger.Self-Service Solutions Applications Documentation

Each Self-Service Solutions Application comes with a comprehensive set of documentation covering the installation, configuration, and use of the product. The application documentation expands and complements the Self-Service Solutions Platform documentation.

The documentation set is divided into the following categories:

- **Manuals**: These manuals cover installing and using the application.
- **Reference**: These reference documents contain information about APIs, databases, configuration files, and so on. These documents are delivered in HTML.

Self-Service Solutions Application Manuals

A full set of these manuals is available for each Self-Service Solutions Application. The documentation set includes the following (in alphabetical order):

- *Business Objects Model Reference*
  Provides a detailed description of the models and entities that make up the Self-Service Solutions Application.
- *Catalog Loader Reference*
  Provides information about the Catalog Loader, including a functional description as well as installation, configuration, and use instructions.
- *Configuration and Development Guide*
  Provides instructions for configuring and developing Self-Service Solutions Application features.
- *Feature Reference*
  Describes the logic and provides use cases for the functional domains of the application.
- *Out-of-the-Box Reference Guide*
  Describes the Self-Service Solutions Application Out-of-the-Box release.
- *Self-Service Installation Guide for Comverse ONE*
  Provides detailed installation, configuration, and deployment instructions for the Self-Service Solutions Application alongside other elements of the Comverse ONE solution.

- *Self-Service Installation and Deployment Guide*
  Provides detailed installation, configuration, and deployment instructions for the Self-Service Solutions Application.
- *Introduction*
  Provides a high-level architectural and functional description of the Self-Service Solutions Application. It covers common features, order management, account management, and bill presentment.

Self-Service Solutions Application References

A full set of these references is available for each Self-Service Solutions Application.The reference documentation set includes the following (in alphabetical order):

- *API Reference*
  Describes usage syntax for the Self-Service Solutions Application APIs. These APIs are used to program the user interface and manage data.
- *Invoice Schema Reference*
  Describes the invoice schema reference of the Self-Service Solutions Application.
- *Presentation Layer Page Flow Reference*
  Describes the page flows of the Self-Service Solutions Application.
- *Specification Entity Relationship Diagrams*
  Provides diagrams describing the actors, use cases, user activity, and storyboard in IBM Rational Rose format.

Self-Service Solutions - Separately Licensed Products

Documentation available with optional, separately-licensed premium products in the Comverse Self-Service Solutions is listed below.

Online Catalog Manager

Online Catalog Manager (OCM) documentation includes the following (in alphabetical order):

- *Introduction to the Online Catalog Manager*
  Provides a high-level architectural and functional description of the Online Catalog Manager.
- *Online Catalog Manager Getting Started Guide*
  Describes the best way to build product catalogs in the Online Catalog Manager. This manual is a template for creating end-user documentation.
- *Online Catalog Manager Installation and Configuration Guide*
  Provides installation and configuration instructions for the Online Catalog Manager.
- *Online Catalog Manager User Documentation Template*
  Describes the use of the Online Catalog Manager. This manual is a template for creating end-user documentation. This manual covers many common concepts and procedures of the OCM.
- *Online Catalog Manager User Guide*
  Provides a detailed description of the concepts and use of the Online Catalog Manager. The topics include:
  - Managing Media Files
  - Managing Offers
  - Managing Prices
  - Managing Products
  - Managing Properties
  - Managing Reference Data
  - Publishing

CSR Portal

The CSR Portal product includes the standard Application documentation, plus the following manual:

- *CSR Portal User Guide*
  A guide to using the CSR Portal UI.

Chapter 1

Network Interfaces Overview

New Features for This Release

The following new features in the Comverse ONE 3.5 TR1.0 release impact the Network Interfaces and Notifications Guide:

- Language Date Format for USSD
- Primary Offer Change

Network Interfaces Overview

The Comverse ONE solution communicates with the entities that provide service to the end user. At a high level, these entities fall into one of the following domains: (1) network or (2) application.

- The network domain includes telecommunication network elements that provide network connectivity to the user. These network elements can be in the traditional Public Switched Telephone Network (PSTN) network (for example, a Mobile Switching Center [MSC] or Service Switching Point [SSP]), or in packet-switched networks (for example, a Serving GPRS Support Node [SGSN], Gateway GPRS Support Node [GGSN], Packet Data Serving Node [PDSN], or Packet Data Gateway [PDG]), or in next generation networks (for example, an IP Multimedia Subsystem [IMS] or Multimedia Domain [MMD]).

  The components in the network domain provide basic services to subscribers such as voice, SMS, or data access. These network elements follow industry-defined interfaces to communicate with the Comverse ONE solution. Supported standards include CAMEL (for GSM and GPRS networks), Wireless Intelligent Network (WIN) (for CDMA networks), and Diameter (for next generation IP networks).
- The application domain includes applications that are provided to the end user. The subscriber accesses these applications via the network.

  Examples of these applications include premium SMS (astrology forecasts or sports scores), content (MP3s, ringtones, wallpapers), and ringback tones. These applications can be delivered by application-specific platforms (for example, Fundial to provide ringback tone service) or general purpose Application Server (AS) or Service Delivery Platforms (SDP) that hosts multiple applications.

  Unlike the network domain, the charging interfaces from the application domain are not well standardized.

The key interfaces for real-time charging supported by the Comverse ONE solution are defined below.

SS7-Based Protocols

The Comverse ONE solution supports a number of SS7-based protocols to facilitate charging from GSM and CDMA networks. These include:

- **CAMEL-2**: Voice charging for GSM networks
- **CAMEL-3**: SMS
- **CAMEL-3**: GPRS
- **WIN (IS-826)**: Voice charging for CDMA networks
- **IS-41P**: Legacy Nortel proprietary protocol for ANSI mobility networks using Nortel MSCs
- **ETSI INAP**: Voice charging for European Telecommunications Standards Institute (ETSI) wireline networks; also used sometimes in mobile networks when the Mobile Switching Center (MSC) supports Intelligent Network Application Part (INAP) protocol
- **INAP CS-1R**: For interfacing with Nortel SSPs

In addition to the SS7 interfaces for rating and charging of activities (voice, SMS, GPRS) that are listed above, the Comverse ONE solution supports several other interfaces. These include:

First, the GSM MAP (Mobile Application Part) interface is supported as follows:

- **MAP SendRoutingInfo Message:** These messages are sent by the Comverse ONE solution to the Home Location Register (HLR) when a call request is received over ISUP or via USSD Callback. The purpose of the message is to obtain subscriber's location from the HLR, which is then used in determining the applicable tariff.
- **USSD:** The Comverse ONE solution supports Unstructured Supplementary Services Data (USSD). USSD is defined by GSM standards in Phase 1 and Phase 2. The following is a list of USSD support in the Comverse ONE solution:
  - **USSD Self-Care (USSD Phase 1 and Phase 2):** The subscriber sends a USSD request to perform various self-care activities, such as checking balances, recharging balances, and so on.
  - **USSD Notifications (USSD Phase 2):** USSD Notifications were introduced in USSD Phase 2 specifications only. The Comverse ONE solution sends USSD notifications to the subscriber. These notifications are described in Chapter 5.
  - **USSD Callback (USSD Phase 1 and Phase 2):** This capability is used in GSM networks to support roaming when no CAMEL-based roaming is available. When the subscriber needs to make a call, she sends a USSD request to the Comverse ONE solution. The request includes the destination number to which the subscriber wishes to make a call. The Comverse ONE solution directs the switch to initiate the calls to the subscriber and the destination, and then connects the two.

Second, the Comverse ONE solution supports Feature Request message of ANSI 41 to provide some self-care capabilities to CDMA subscribers.

Third, the Comverse ONE solution supports SMPP 3.4 to send SMS-based notifications to subscribers.

Session Initiation Protocol (SIP)

SIP is a signaling protocol used for setting up and tearing down multimedia communication sessions such as voice and video calls in the IP network. SIP allows Comverse ONE to perform real time charging of calls originating or terminating in SIP signaling-based networks. Current support is for voice charging of a two-party call including precall announcements.

The SIP protocol is an IP-based Application Layer protocol. It is a text-based protocol, sharing many elements of the HTTP upon which it is based, allowing for easy inspection by administrators.

SIP does not have a real time charging interface defined. In order to provide real time charging of SIP calls, Comverse ONE acts as a SIP back-to-back user agent (B2BUA). A B2BUA is a logical SIP network element. It resides between both end points of a phone call or communication session and divides the communication session into two call legs and mediates all SIP signaling between both ends of the call, from call establishment to termination. Each call is tracked from beginning to end, allowing the operators of the B2BUA to offer value-added features to the call.

Comverse ONE acts as a B2BUA in the following networks:

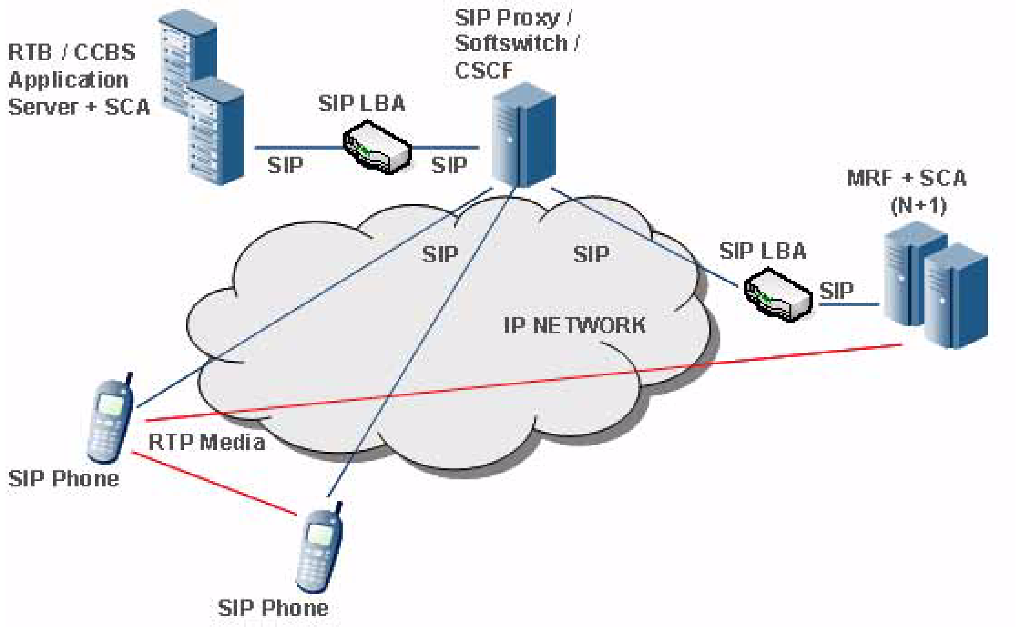
- Class 5 VOIP
- Class 4 backhaul
- IMS

In the originating call leg, the B2BUA acts as a user agent server (UAS) and processes the request as a user agent client (UAC) to the destination end, handling the signaling between end points back-to-back. A B2BUA maintains complete state for the calls it handles. B2BUA can also manage the call and decide to perform a number of actions including disconnecting a call (that is running out of prepaid credit), forward a call to customer service or connect a call to an announcement server.

A B2BUA server initially appears to be a proxy server. Like a proxy server, it can connect a point-to-point call using its routing capabilities. Unlike a proxy server, the B2BUA performs additional tasks such as disconnecting an active call.

The following figure shows the overall architecture of SIP.

Figure 1 SIP High Level Design Overview



The Comverse ONE/RTB/CCBS application servers are connected via a SIP Load Balancer (LBA) to the SIP proxy/softswitch/CSCF. The Media Resource Function (MRF) is also connected via a SEP LBA to the SIP proxy/softswitch/CSCF.

The Signaling Control Agent (SCA) is a Comverse component that provides signaling access for Comverse applications requiring a signaling interface to basic VoIP and/or IMS networks. The SCA supports signaling based on SIP, which is currently the main signaling protocol in VoIP networks.

SIP Service Logic Unit (SLU)

The SIP SLU is the Comverse ONE/RTB/CCBS application server.

SIP Media Resource Function / Comverse Media Server (CMS)

CMS provides the SIP MRF. It uses SCA for SIP access. The SIP SLU communicates with CMS to request media services such as announcements.

SIP Load Balancing and Redundancy

The SCA is a standalone network entity and does not handle load balancing or redundancy issues. Instead, the SCA relies on an external entity for this. This entity can be any SIP-aware load balancer (SLB).

SEE See the Product Catalog User Guide for details regarding SIP configuration.

Chapter 2

CAMEL Enabled Services

CAMEL Enabled Services Overview

This chapter describes the specifications of the Circuit Switched Fax/Data Bearer feature as well as information for the CAMEL 3 Short Message Service (SMS) feature . An overview of CAMEL 3 GPRS and information is also included.

NOTE Refer to the Product Catalog Users Guide for details regarding CAMEL Enabled Services provisioning.

Circuit Switched Fax / Data Bearer

The Circuit Switched Fax/Data Bearer feature enables an operator to charge different rates for telephony, fax, and other data services. This feature is initially implemented for the IN networks using CAMEL messaging.

The type of call (telephony, fax, or other data services) is determined from the highLayerCompatibility field in the InitialDP message and the provisioned rate is applied to the call. The highLayerCompatibility field is an optional field in the European Telecommunications Standards Institute (ETSI) CAMEL, and the network switches must pass this information for the feature to work. If this field is not passed, the default bearer type of telephony is assumed. The coding of the highLayerCompatibility field is specified in ITU Recommendation Q.931 (ISDN) 05/98.

The following transaction types are uniquely identified and charged:

- Telephony (default)
- FAX_GRP_2_3
- FAX_GRP_4
- SYNTAX_VIDEOTEX
- INT_VIDEOTEX
- INTTELEX
- X400

A BearerType field is included in the CDRs and the call_type field in the Call History record has values that reflect the setting of the highLayerCompatibility field. In a future release, similar information is extracted from the ISUP IAM message using the User teleservice information field (high layer compatibility IE contained in access transport, 2nd priority). The CDR/call history formats are updated for a CAMEL (SGU)-based system.

NOTE Refer to the Rating Technical Reference for additional details regarding CDRs and usage records.

CAMEL 3 SMS

Support for billing of originating SMS services is built directly into the CAMEL 3 protocol, enabling an SCP (such as the Comverse ONE solution) to allow/disallow individual SMS transactions for each subscriber. The Comverse ONE solution includes support for billing originating SMS services over CAMEL 3, including support for location-based billing.

For location-based billing purposes, CAMEL Phase 3 (CAP3) SMS calls use the LBB_precedence file to determine the locations. The LBB_precedence file supports CAP3 SMS billing by supporting new entries for CAP3 SMS.

Mapping of the SMS call type is done through the Activity Definition table, as is currently done for MO-SMS support via the Event Charging Interface. There is no difference in the entries for CAP3 and those for Event Charging Interface SMS support.

If there is no MO-SMS Bearer Application subtype defined for the original call type, the SMS is not allowed. CAP3 SMSs are treated like a call for lifecycle purposes, with the exception of forwarding the call (to the Recharge Server or Customer Center). A subscriber in Fraud Lockout state is not allowed to send SMSs.

SMS Center (SMSC)

When an SMS message is sent, it is routed through SMSCs before reaching its destination. SMSCs developed by different vendors use different communication protocols. Two SMSCs cannot be connected if they do not support a common protocol. An SMS gateway resolves this problem by acting as a protocol translator, enabling the connection between disparate SMSCs that otherwise could not interact.

In the Comverse ONE solution, the open-source Kannel SMS gateway is integrated into the Unified Platform.

SEE

Refer to the Unified Platform Guide for further details regarding Kannel SMS gateway.

Reservations

For one CAMEL 3 SMS activity, there is only one reservation issued by URE_SMS, an SLU service application. When the Subscriber Data Server (SDS) calculates the reservation/consumption for a CAMEL 3 SMS activity, it always uses the first consumption charge of the primary tariff. The reservation is either granted or rejected and the reservation chunk size is always one SMS. There is no such thing as half an SMS or two SMSs since the CAMEL 3 protocol only processes one SMS at a time.

CAMEL 3 GPRS

The Comverse ONE solution supports billing for Third Generation (3G) network packet data services using the CAMEL 3 General Packet Radio Services (GPRS) in GSM networks. GSM data services vary from streaming media to Internet Web surfing. These data services are fundamentally limited only by mobile handset capabilities and interface bandwidth.

The CAMEL 3 (CAP Phase 3) protocol has support for prepaid data services billing built directly into the protocol so that an SCP (such as the Comverse ONE solution) can allow or disallow and charge for data transactions. The Comverse ONE solution receives usage information for subscriber services in the form of octets or seconds from the GPRS network. The Comverse ONE solution rates this usage by converting it into corresponding currency units such as dollars.

The reservation-based mechanism is used for charging. A certain amount of usage is authorized and the corresponding balances are set aside when the network requests a reservation. The balances are permanently reduced only after consumption is confirmed. If a part of the reservation is not used, it is returned and the balances are readjusted accordingly. Reservation amounts are configurable.

Pricing

The Comverse ONE solution supports pricing for CAMEL 3 data services using a highly configurable, flexible, reservations-based pricing scheme. This allows support for data pricing by:

- Access Point Name (APN)
- Application
- Volume (octets)
- Duration (seconds of connection)
- Combination of Volume and Duration
- Quality of Service

Standard billing features are applied to data pricing and support billing options such as:

- Different rates based upon date, time, location, and subscriber primary offer
- Concurrent tariffs
- Telescoping charges
- Grace period
- Settlement tariffs (This is used to designate one of the concurrent tariffs as a settlement tariff. The tariff is not charged against the subscriber, but is reflected in the CDR if the field is selected. This information is used for processing by a downstream billing system.)
- Free octets and/or seconds
- Bonuses and discounts based on usage (optional feature)

Data Location Relations

In CAP 3, the APN identifies the particular data service used and the requirement is to charge different APNs at different rates. The characteristics of the data connection (Quality of Service) are identified in CAMEL 3, and the Comverse ONE solution is able to charge different rates based on this information. The Quality of Service refers to the capability of a network to provide better service to selected network traffic. It is identified at the beginning of each session and used for tariffing. In addition, each APN is charged based on duration and/or volume, and this is configurable.

Since the InitialDPGPRS message does not contain a destination number, the billing engine assumes a destination of anywhere for all of the data services. Friends & Family is not supported for data charging.

Data Tariffs

The CAMEL 3 data services are billed by byte usage and/or duration (time). The Comverse ONE solution billing infrastructure is flexible and has configuration to support any combination of the two.

Reservation Mechanism Billing

The Reservation/Balance Nibbling scheme for charging the subscriber is supported by this interface. Each data reservation consists of an amount of money (a nibble from the subscriber’s available balances), and an expiration time, indicating for how long the reservation is used.

These two attributes are part of the Apply Charging message sent by the Comverse ONE solution to tell the switch Gateway GPRS Support Node (GGSN) how much activity the subscriber is allowed before it reports back to the SCP (the Comverse ONE solution).

For example, a reservation of 20 Kbytes and 60 minutes means that the GGSN must report back in either 60 minutes, or when the subscriber consumes 20 Kbytes, whichever comes first.

Example 1

In this example, the subscriber has $10 available in balance, 0 free bytes, and attempts to start a browser session.

1. The switch sends a message to the Comverse ONE solution indicating that this subscriber is attempting to browse the Web, and requests a reservation. The Comverse ONE solution analyzes the request, and based upon primary offer parameters, determines that a Web browsing reservation for this subscriber at this date/time, consists of 100 Kbytes, with a four hour expiration period.
2. The Comverse ONE solution determines that the subscriber has no free bytes available, so the balance is used. The Comverse ONE solution then calculates the charge for this reservation of 100 Kbytes (assume $2), and reserves this amount from the subscriber’s balance. While the subscriber has $10 in balance, only $8 is available for other services.
3. A reservation is then sent back to the switch, indicating that the subscriber has 100 Kbytes and four hours, and that the switch must report back when the 100 Kbytes is used, or after four hours, whichever comes first.
4. Assume that after four hours, the subscriber has only used 50 Kbytes. At that time, the switch reports back that the subscriber has used 50 Kbytes of the reservation.
5. The Comverse ONE solution calculates the value of the 50K unused bytes (assume $1). The Comverse ONE solution then debits the subscriber’s account by $1 and cancels the reservation. The subscriber now has a balance of $9, of which all is available. Activity histories and (optionally) CDRs are generated at this time.
6. If the subscriber continues to surf the Web, Steps 1 – 5 are repeated.

Example 2

In this example, the subscriber has $10 available in balance, 200K free bytes, and attempts to start a browser session.

1. The switch sends a message to the Comverse ONE solution indicating that this subscriber is attempting to browse the Web and requests a reservation. The Comverse ONE solution analyzes the request, and based upon primary offer parameters, determines that a Web browsing reservation for this subscriber at this date/time, consists of 100 Kbytes, with a four hour expiration period.
2. The Comverse ONE solution determines that the subscriber can use free bytes to cover this reservation and reserves 100 Kbytes from the subscriber’s free bytes. While the subscriber has 200 Kbytes in his free bytes balance. Now only 100 Kbytes is available for other services.
3. A reservation is then sent back to the switch, indicating that the subscriber has 100 Kbytes and four hours, and that the switch must report back when the 100 Kbytes is used, or after four hours, whichever comes first.
4. Assume that after four hours, the subscriber has only used 50 Kbytes. At that time, the switch reports back that the subscriber has used 50 Kbytes of the reservation.
5. The Comverse ONE solution then debits the subscriber’s free byte balance account by 50 Kbytes and cancels the reservation. The subscriber now has a free byte balance of 150 Kbytes, of which all is available. Activity histories and (optionally) CDRs are generated at this time.
6. If the subscriber continues to surf the Web, steps 1 – 5 are repeated.

Current GPRS Mappings

NOTE It is assumed that the applicable Comverse ONE solution hardware and software provisioning to configure the Service Logic Unit (SLU) and the Comverse ONE solution database is completed by tier 4 support.

NOTE Refer to the Product Catalog User Guide for specific provisioning steps.

For each mapping in the selection list, the following information fields are provided:

GPRS Parameters:

- **Access Point:** Contains the APN used when passing the PDP context and identifies the network type and particular data service used. Up to 100-character string format with labels separated by dots, for example, Label1.Label2.Label3.
- **Quality of Service:** Quality of Service refers to the capability of a network to provide better service to selected network traffic and represents the characteristics of the data connection. The Quality of Service identified at the beginning of the session is used for tariffing, and the Comverse ONE solution charges different rates based on this information. A Quality of Service parameter is a string of hex digits (0–9, A–F). Quality of Service parameters are contained in a table maintained by the network administrator.

Activity Parameters:

- **Application:** Name of the application for which usage charges are determined as defined in the Activity Definition table.
- **Subtype:** Subtype name associated with this application activity as defined in the Activity Definition table.
- **Charge Type:** Specifies if the Comverse ONE solution charges for usage duration (seconds), volume (octets), or both.

Provision GPRS Mapping

In order to rate transactions over the GPRS external interface, the following provisioning procedures are completed for GPRS mappings:

1. Define Location Relations for GPRS charging. This procedure is optional and is used for location-dependent GPRS charging.
2. Define a Pseudo Location for all GPRS B party (terminating) locations. This is required for step 3.
3. Define the GPRS environment variable in the SLU.
4. Define the necessary GPRS application, subtype, and unit type activity combinations needed to charge for the activity and different levels of the service, if applicable.
5. Map the external service network activity and Quality of Service to the predefined Comverse ONE solution activity and charging method.

Define the GPRS Environmental Variable

NOTE This procedure must be completed before the OMNI platform can be brought up. This procedure must be performed by Comverse Tier 4 support.

1. Verify that a Pseudo Location is defined for all GPRS B party (terminating) locations.

2. In the database, locate the 16-digit location indicator (LI) associated with the GPRS Pseudo Location defined in step 1.
3. In the SLU where the GPRS application runs, set the new GPRS environmental variable to **GPRS_Blocation** in the omnisetup file.

Provisioning Examples

This section provides three examples of how the Comverse ONE solution is provisioned to charge for different GPRS services:

- Charging by volume
- Charging by duration
- Charging by duration and volume

In order to rate transactions over the GPRS external interface, the following provisioning procedures are completed:

- Defining Location Relations for GPRS charging (optional)
- Defining the necessary GPRS application, subtype, and unit type activity combinations needed to charge for the activity and different levels of the service, if applicable.
- Mapping the external service network activity and Quality of Service to the predefined Comverse ONE solution activity and charging method.

Charging by Volume

In this example, the network operator wants to make EarthLink Services available to subscribers using the GPRS network. EarthLink transactions are charged by volume and there are two levels of service quality: premium and economy.

NOTE Refer to the appropriate Quality of Service table of values to determine the Quality of Service for the Economy Quality of Service.

Charging by Duration

In this example, the network operator wants to make AOL Services available to subscribers using the GPRS network. AOL transactions are charged by duration and there are three levels of service quality: economy, standard, and premium.

Charging by Duration and Volume

In this example, the network operator wants to make Earthlink services available to subscribers using the GPRS network. Charging for Earthlink transactions is by both duration and volume for premium service.

When configured for charging by both duration and volume, the Comverse ONE solution performs these charging operations concurrently for the transaction.

Chapter 3

USSD Enabled Services

USSD Enabled Services Overview

This chapter provides information for provisioning two Comverse ONE solution features that are based on the Unstructured Supplementary Services Data (USSD) protocol: USSD Callback and USSD Recharge and Information Server.

USSD Callback

The USSD Callback feature enables the Comverse ONE solution to support subscribers roaming in Global System for Mobile Communication (GSM) networks that are not compliant with the CAMEL Phase 2 standard.

To support roaming in a prepaid environment, each call attempt is individually authorized and active calls are monitored. This requires the involvement of the Comverse ONE solution during both call setup and call progress.

The most convenient method of supporting prepaid roaming uses the GSM CAMEL Phase 2 standard. This Intelligent Network (IN) call model allows a subscriber to roam in any network that is CAMEL compliant. Unfortunately, not all GSM networks around the world are compliant, and a backup approach is required to support roaming in these cases. One of the more popular alternative methods is based on USSD callback.

The Comverse ONE solution complies with the following USSD standards:

- USSD Standard Phase 1 and 2
- Ericsson USSD Phase 1 and 2 (not based on the GSM Mobile Application Part (MAP) J interface specification)
- Nokia Phase 1 and 2 (variant of the standard)

Call Placement

In order to place an outgoing call while roaming, the subscriber sends a USSD message with two essential parameters:

- The service code defined for callback
- The destination number to be dialed

The service code is a 2- or 3-digit number defined by the local carrier. It is used to determine where the USSD message is sent. The subscriber’s Home Location Register (HLR) is configured to recognize the service code and deliver the message to the Comverse ONE solution.

The destination number is a valid telephone number as defined relative to the subscriber’s home zone. The Comverse ONE solution connects the call assuming numbering plan rules appropriate to the subscriber’s home zone (regardless of the subscriber’s actual location).

Consistent with rules defined by the standards bodies, a number of * and # key-presses are required with the service code and destination number. For example, if the service code is defined as 105 and the number dialed is 6091234567, the subscriber enters the following sequence of characters and then presses Send:

##105*6091234567#

Upon receiving this message from the subscriber, the Comverse ONE solution first identifies the subscriber. Depending on the specific capabilities of the local and roaming networks, the subscriber account is recognized using either the Mobile Switch Integrated Services Digital Network (MSISDN) or the International Mobile Subscriber Identifier (IMSI). Next, the Comverse ONE solution determines if the subscriber is authorized to make roaming calls and if sufficient credit is available in the account to complete the call. If the available balance is insufficient, a USSD response informs the subscriber of the call denial. If sufficient credit is available, the Comverse ONE solution immediately places a call back to the subscriber. When the call is answered, a brief message is played to the subscriber (using their default language) to confirm that this is the requested callback.

For example, the subscriber hears the message:

“This is your callback. Please hold while we connect you.”

After the subscriber answers the call, the Comverse ONE solution sends the requested call to the destination number. The timing associated with call charges is adjusted and two options are available:

- The call is chargeable when the subscriber answers the callback, whether or not the called party is reached.
- The call is chargeable only if both calls are answered. In this case, the subscriber is not charged if the called party is not reachable. The call duration is set to start from either the subscriber or the destination party answering the call.

A tariff is selected for the call and the maximum call duration is calculated relative to the amount of prepaid credit in the subscriber’s account. The Comverse ONE solution ensures that the cost of the call does not exceed the amount of prepaid credit available. If the available balance is consumed, the prepaid call is disconnected in real-time. The monitoring and control of a prepaid call placed by a roaming subscriber is the same as the monitoring and control of any other type of prepaid call.

USSD Call Charging

The full array of charging options offered by the Comverse ONE solution applies to calls placed using the USSD callback mechanism. As part of call setup, the Comverse ONE solution issues a MAP query to the HLR to determine the subscriber’s location. The location relation is determined by noting both the subscriber’s location and the location of the called party.

All charging features available to the carrier when specifying tariffs for nonroaming subscribers are also applicable to roaming subscribers. For example, the call is charged based on:

- The date and time of the call (relative to the subscriber’s home zone).
- The location of the called party (usually implied from the sequence of digits dialed).
- Whether or not the call is placed to a member of the subscriber’s Friends & Family list.
- Any allotment of free seconds associated with the subscriber’s primary offer.

Since the Comverse ONE solution allows more than one tariff to be applied concurrently to the same call, it is possible for the service provider to charge a roaming fee in addition to the usual charge for a call. For example, the tariff for a call placed when roaming is the same as the tariff applied when calling from home, supplemented by an agreed roaming charge (which may also depend on the duration of the call).

A successful USSD Callback call generates two Call History records: one for the USSD call, and one for the voice call, as well as two Call Detail Records (CDRs): one for the callback, and the other for the call to the destination. The outbound legs of the two CDRs are correlated by the call reference and associated call reference fields provided that CDRs are configured via the CDRFormat.config file to include the REFERENCE and ASSOC_CALL_REF fields.

NOTE The Follow-On Call and Virtual Private Network (VPN) features do not work for USSD Callback operations.

USSD Callback Enhancement

USSD Callback calls are distinguished as On-net or Off-net to allow preferential rating.

- **On-net:** Both A party and B party belong to the same operator.
- **Off-net:** A party calls B party who belongs to a different operator.

The Mobile Number Portability (MNP) solution is used to differentiate On-net and Off-net calls. The Comverse ONE solution uses Service Control Point (SCP) capabilities to initiate ISDN User Part (ISUP) calls to both calling and called parties and then loops the two calls. MNP provides a specific prefix to each number which is also defined as a location for the operator. The prefix allows the operator to identify both parties as subscribers or nonsubscribers and rate the call accordingly.

USSD Callback Provisioning

To provision USSD Callback, log in to the Product Catalog (PC), and perform the following procedures:

- Setup Service – USSD
- Billing Model Provisioning – Locations – Location Relations
- Response Message Provisioning – Primary Offer – USSD and USSD Callback

NOTE Refer to the Product Catalog Users Guide for details on USSD provisioning.

NOTE The HLR must be configured to recognize subscriber service codes. Contact the communications network administrator.

USSD Recharge and Information Server

The USSD server provides a recharge and information service to subscribers without the use of IVR sessions that have the potential to overload the network. By dialing configurable service codes or information strings, subscribers retrieve information about accounts via text messages on their handsets or recharge accounts using a recharge voucher.

USSD transmits information or instructions over the Global System for Mobile Communication (GSM) network in a manner similar to that of the Short Message Service (SMS). Unlike SMS, USSD is session-oriented and does not store and forward the information.

When a user accesses a USSD service, a session is established and the radio connection stays open until the user or application terminates it. USSD commands are routed back to the home mobile network's Home Location Register (HLR) and work just as well when users are roaming.

When the service code entered by the subscriber is interpreted and validated, the system is able to retrieve account information such as balance and expiration date, or allow the subscriber to recharge an account balance using a recharge voucher.

Date Format

The USSD Recharge and Information Server had previously displayed the balance expiration date as YYYY-MM-DD HH:MM:SS. Going forward, balance expiration will show the date only and no time stamp.

- For subscribers with French notification language, the date format will be displayed as DD/MM/YYYY.
- For subscribers with English notification language, the date format will be displayed as MM/DD/YYY.

- For subscribers with notification languages other than English or French, the default format specified in the XML (ussd_cfg.xml) file will be used.

If the default format is not specified in the configuration XML file, then the date format will be the default Oracle format as it will be stored in the database.

USSD Recharge Provisioning

To provision the USSD Recharge and Information Server feature, log in to the Product Catalog (PC), and perform the following procedures:

- Setup Service – USSD
- Billing Model Provisioning – Locations – Location Relations
- Response Message Provisioning – Primary Offer – USSD

NOTE

The applicable HLR(s) must be configured so that the service codes are identified and to allow forwarding of any Mobile Application Part (MAP) messages with the service codes to the Comverse ONE solution. Contact the communications network administrator.

USSD Liability Redirection Override

There are occasions where a subscriber needs a way to override the existing Liability Redirection rules and force the charges to a specific target. Liability Redirection Override provides this option.

Example 1: Target Account / Shadow Subscriber has no available funds

A child has her own account with all of her local usage charges redirected to her parent’s account. However, the parent forgot to top-up the account and it is now empty. Thus the child's local usage is blocked. The child chooses to override the redirection to the parent's account and pay for the usage herself.

Example 2: Usage exceptions

Nick has an arrangement where all of his calls during normal working hours are redirected to his employer's account. However, Nick wants any personal calls made during working hours charged to his personal account.

A variation of this is that due to an emergency Nick is called into work over the weekend and wants all of his business-related calls to be charged to his employer.

Example 3: Privacy concerns

Patty has an arrangement with her employer that he pays for all of her local calls. However, she is interviewing for another job locally with a competitor and doesn't want her employer to see that she has called the competitor.

While there is no reason to deny a subscriber the capability to override Liability Redirection rules and charge the usage to himself, there must be some control regarding a subscriber redirecting usage elsewhere.

Liability Redirection Override Management

At the high level, a subscriber can override the provisioned Liability Redirection rules to force Liability Redirection to himself or to a Shadow Subscriber. A subscriber cannot override the Liability Redirection rules to force the Liability to be redirected to a specific account.

The owner of the Shadow Subscriber can configure whether or not the Shadow Subscriber accepts Liability Redirection Overrides.

The Shadow Subscriber allows Liability Redirection Overrides but cannot limit what usage charges are overridden (other than by the assigned usage plans and available funds).

The using subscriber has a parameter called LR_Override, which controls this functionality.

LR_Override can be off, indicating that the configured Liability Redirection rules should be used, or set to a value to indicate that all usage charges should be directed to himself, or to a Shadow Subscriber.

The subscriber sets the LR_Override value via USSD, FeatureReq, CSM, or Self-Care. The request is initiated with an access number along with a new Liability Redirection override ID. For example:

`#123*3#` - 123 is the access number and the final 3 is the new Liability Redirection ID.

Valid Liability Redirection override values are requested by an access number. For example:

`#456#` - 456 is the access number.

When the value is set, all usage is charged to the specified target (assuming the target allows the usage and has enough funds) until the subscriber changes the value to something else.

Each IN-enabled real and Shadow Subscriber has a configurable single-digit value that is used as the LR_Override_Code with 0 reserved for no override and 1 reserved for the subscriber himself.

There is also an associated text field that is used when displaying information related to the LR_Override values.

For example:

- Set LR_Override to 0 indicates that the provisioned Liability Redirection rules should be used.
- Set LR_Override to 1 indicates that all usage charges should be directed to himself.
- Set LR_Override to 3 indicates that all usage charges should be directed to Shadow Subscriber1.
- Set LR_Override to 5 indicates that all usage charges should be directed to Shadow Subscriber2.

Liability Redirection Override Limitations

- Liability Redirection Override is only available for IN-enabled subscribers.
- Liability Redirection Override is only applicable to usage charges.

Liability Redirection Override Provisioning

Liability Redirection Override provisioning requires changes in the Product Catalog to set up the messages and in the Customer Center to choose the account for liability transfer.

Using the Product Catalog:

- At the global screen, set the USSD codes for the global response.
- Within the PO, set the response message.

Using the Customer Center:

- On the account menu, click Liability Transfer.
- Locate the LR for modification and choose the target or source ID.

NOTE Refer to the Product Catalog User Guide for details on USSD code provisioning regarding Liability Redirection Override response messages.

NOTE Refer to the Customer Center User Guide for details on Liability Redirection Override (Liability Transfer) provisioning.

Chapter 4

Platform Architecture Overview

Platform Architecture Overview

The Comverse ONE solution is a flexible technology capable of adapting to the needs of the local network environment. The platform integrates with all the major switch vendors and supports a variety of Intelligent Network (IN) signaling technologies. The same platform supports a mixed vendor environment, as well as mixed signaling interfaces including IN interfaces and Pre-IN concurrently.

Comverse ONE Solution Protocols

The Comverse ONE solution platform supports a variety of protocols, including:

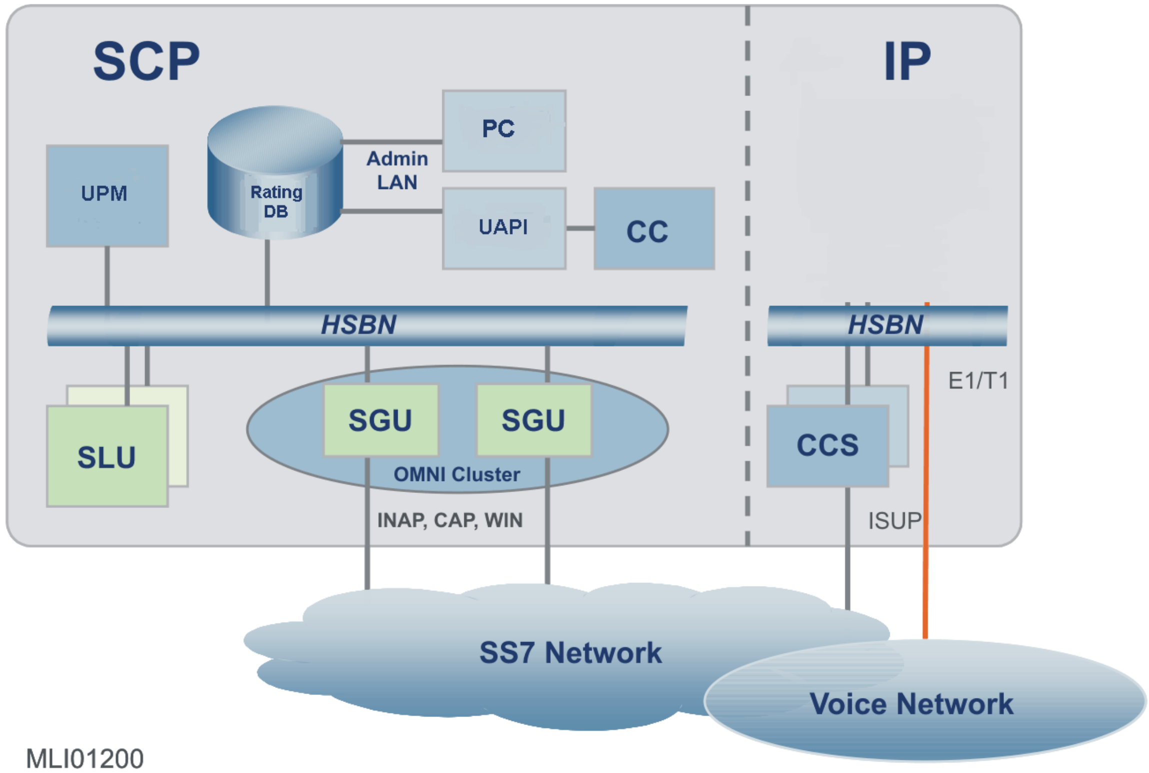
- Customized Applications for Mobile network Enhanced Logic (CAMEL) Phase 2
- CAMEL Phase 3
- Wireless Intelligent Network (WIN) Phase 2 (IS.826)
- ISDN User Part (ISUP)
- Intelligent Network Application Protocol (INAP) CS-1

A traditional pre-IN approach, based on the processing of regular Signaling System 7 (SS7) ISDN User Part (ISUP) messages consumes ports at a high rate at the switch. An IN approach reduces this resource burden in the network. Comverse ONE solution operates in Pre-IN mode, in full IN mode, and supports both modes of operation simultaneously. The same system hardware, billing application, service, and subscriber database are used in each case, and (with minor exceptions) are unaffected by a change in signaling configuration.

IN Configuration

The IN Protocol Application is responsible for all protocol processing above the SS7 SCCP protocol layer. Call processing logic at the Mobile Switching Center (MSC) is interrupted temporarily at certain predefined points, and an IN-specific signaling message (for example, an INAP message) is sent to the Comverse ONE solution platform. As a result, service logic is invoked at the platform. A check is made to ensure that the subscriber has sufficient funds available to make the call and an IN response message is returned to the MSC to direct completion of the call. This call setup scenario is shown in Comverse ONE Solution IN Configuration Architecture.

Figure 2 Comverse ONE Solution IN Configuration Architecture



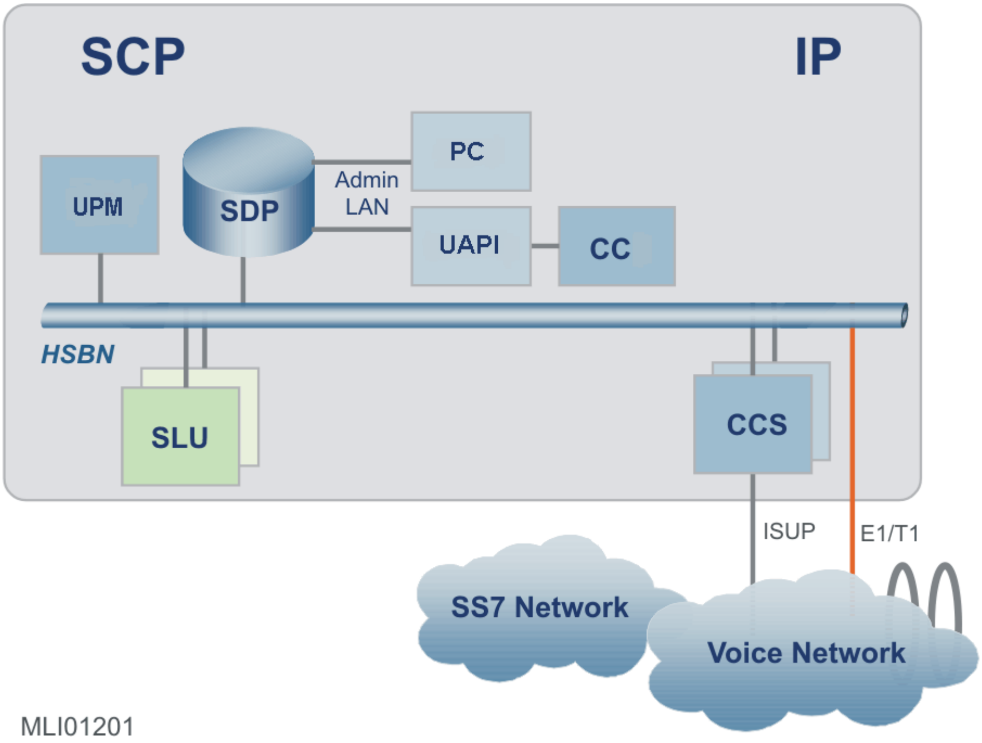
MLI01200

IN affects the protocol used to support the normal mechanics of telephony. Some examples of IN protocols include: INAP, CAMEL, WIN, and TCAP (Transaction Capabilities Application Part). The SS7 stack is comprised of these protocols in an IN system and is located on the Service Logic Unit (SLU) in the Comverse ONE solution.

Pre-IN Configuration

A Pre-IN approach to the Comverse ONE solution is based on the processing of regular SS7 ISUP messages during call setup. Calls requiring the Comverse ONE solution treatment are recognized in the network and diverted along a specific trunk group. The selected trunk does not terminate at the Comverse ONE solution platform but is paired to another trunk at the switch. No trunks are required between Comverse ONE solution and the MSC. Only SS7 signaling information is passed to the platform (typically along an assigned E1 channel) from the MSC. The Comverse ONE solution application logic is invoked as a result of the incoming signaling message and a check is made to ensure that the subscriber has sufficient funds to make the call. Call setup messages are returned to the switch to direct completion of the call using the partner port in the trunk pair. This call setup scenario is shown in the following workflow.

Figure 3 Comverse ONE Solution Pre-IN Configuration Architecture



Component Overview

The following sections describe the various components of the Comverse ONE solution.

Service Logic Unit (SLU)

The SLU is the main point of control for the Comverse ONE solution. The SLU directs and coordinates the behavior of resources at other components of the system. The SLU consists of two distinct parts:

- The Unified Rating Engine (URE) provides the environment in which real-time application logic executes. Platform functions are presented to resident applications in a generic and seamless way and are incorporated in logic programs in the form of library routines.
- The Intelligent Peripheral (IP) Core Function manages voice and switching resources at the platform, converting generic requests from the billing application into specific commands that affect specific resources.

The URE and the IP Core Function are both implemented using OMNI software technology. The OMNI software platform provides an environment that is scaleable, distributed, and which protects against a variety of hardware and software failures. The SLU runs on a single board computer known as the Dual Processor Module (DPM) in a CompactPCI back-plane. Depending on the capacity needs of the system, several SLUs are included in a system configuration. An N+1 redundancy scheme is adopted where one more SLU is included in the system than is needed to handle the anticipated traffic load. As an option, the system is configured with 2N redundancy, where a hot stand-by is associated with every active SLU processor.

Several interfaces require separate SLU pools dedicated to a specific function. These are:

- Call Control Server (CCS)
- General Packet Radio Services (GPRS) SLU
- Open Services Access (OSA) SLU

- Diameter SLU

NOTE Event Charging Interface, GPRS, OSA, and Diameter are value-added options that are available with the Comverse ONE solution.

Call Control Server (CCS)

The CCS is responsible for processing SS7 ISUP messages associated with IVR sessions. It supports all layers of the SS7 protocol (including MTP and ISUP) as defined by a variety of international standardization bodies, including International Telecommunications Union (ITU) and American National Standards Institute (ANSI). Over 30 national variations of the ISUP standards are available from Comverse.

The CCS takes advantage of OMNI middleware platform, in particular its ability to distribute signaling software across multiple computers while maintaining a single system image (a single point code) to the network. Each computer shares the processing load, assuming responsibility for all call control functions in the event of hardware or software failure at its peer. The CCS supports up to eight signaling links at each computer. A maximum of 16 links per link set is supported and up to 8 link sets per route set. The CCS runs on SLU hardware with the SS7 software running on OMNI middleware. Depending on capacity needs, several CCS units are included in a single platform.

Signaling Gateway Unit (SGU)

The SGU is the IN message handler. This unit shares an architecture very similar to the CCS. The SGU processes very large numbers of signaling messages (such as CAP-2 or CAP-3 messages) while presenting a single point code to the network. Refer to the *Signaling Gateway Unit* manual for additional details.

Unified Platform Manager (UPM)

The UPM is the central management point for the Comverse ONE solution platform. UPM offers a single, consistent system image to all external management agents. The UPM controls access to all management and support functions, offering facilities to both system administrators and external legacy systems (via an administrative LAN). UPM provides storage for Call Details Records (CDRs) and outage records and serves as external host for billing collection.

High Speed Backbone Network (HSBN)

The HSBN is a critical component of the Comverse ONE solution because it allows internal platform components to communicate in real-time. Its dual backbone design is based around a segmented Ethernet hub. Up to eight separate network segments, each capable of 100Mb/s, coupled with a powerful 1.2Gb/s backplane, deliver the required bandwidth. The HSBN design ensures high availability and no single point of failure. Platform components are attached to both backbones, allowing them to communicate even in the case of network failure.

Administrative Local Area Network (LAN)

The Admin LAN transfers administrative data and commands via TCP/IP. The Admin LAN is implemented on the same switching hub used for the HSBN, but is totally isolated from the HSBN.

Network Switches

In order to provide service, the Comverse ONE solution must be connected to one or several network switches. In some cases, the Comverse ONE solution is connected to several networks, each one through one or more switches.

Each network switch connected to the Comverse ONE solution has a Signaling Point Code (SPC). An SPC is an address in the SS7 network. The Comverse ONE solution has its own SPC.

NOTE The Comverse ONE solution has multiple point codes if the system is very large.

Trunk Groups (TG)

Each switch connected to CMS 3 has multiple voice channels (usually several hundred channels) that are known to the Comverse ONE solution and the switch. These channels are organized in the form of TGs. A TG is any collection of voice channels that serve a given purpose.

Each switch is connected to the Comverse ONE solution through one or several TGs. For instance, two TGs are used, one for incoming calls and one for outgoing calls. The exact number of TGs depends on the details of the specific site and on how calls are routed to and from the Comverse ONE solution.

This general definition of TGs is in conflict with some of the uses of TGs elsewhere in the Comverse System, mainly on the CCS. The CCS is different from standard TGs in that each E1/T1 line (such as DTM) is a separate TG.

TGs do not have to be defined using full E1/T1 lines. The voice channels carried on an E1 line can be split between TGs.

Circuit Identification Codes (CIC)

Each voice channel between a given network switch and the Comverse ONE solution has a unique identification code known as the CIC. There is no standard convention of how the codes are assigned. All 32 channels of an E1 line (including the D channels) are assigned CICs.

The CIC uniquely identifies a specific channel connected to a given switch. However, it is not unique across the interfaces with several switches. Only the combination of a switch name (or DPC) and a CIC uniquely identify a voice channel.

From a network perspective, TGs are defined by specifying which CICs belong to each TG. In order to achieve proper operation, this must be consistent on the network switch and the Comverse ONE solution.

SS7 Signaling

SS7 signaling links carry all control messages between the Comverse ONE solution and the switches to which it is connected. The signaling links do not have to be physically connected to the same switches they support. With SS7, there is no necessary relationship between the physical connection of the voice circuits and the physical connection of the SS7 links.

For instance, the links are connected through a network Signaling Transfer Point (STP), which acts as a router for SS7 messages and forwards them to the appropriate switches, based on the SPC number used in each message. In this case, the links are normally connected via V.35 lines directly to the STP.

Another alternative is to have the links connected directly to the switches through some of the D channels of the E1 lines (or T1 lines) connected to that switch. There is no need to use every E1 D channel because a single link over a single D channel supports dozens of E1 lines (hundreds of ports).

There are hybrids where one switch also performs an STP function for other switches, so that links connected to that switch also serve TGs connected between the Comverse ONE solution and other switches. The switch with the STP function handles its own messages (identified by the SPC) and routes other messages to other switches (also based on the SPC).

There is no mandatory one-to-one correlation between voice channels connected to specific switches and the physical connection of links supporting these voice channels. The only requirement is to have a well-defined path (links) to send SS7 messages to every switch. The SS7 network manages to route messages sent on the proper links to the right destination.

Redundancy

The Comverse ONE solution is fully redundant. No single point of hardware or software failure results in unavailability of the platform. Further, the platform continues processing even when certain combinations of failure occur. Redundancy is built into every component of the system:

- An N+1 redundancy scheme protects against SLU processor failure. One more SLU is deployed than is necessary to handle the anticipated processing load. In the event of SLU failure, no stable calls are dropped and processing at other SLUs is unaffected. As an option, SLUs are configured with 2N redundancy, whereby a hot stand-by is associated with every active SLU processor. The 2N configuration is recommended for Pre-IN configurations. In the event of failure of the primary processor, the backup processor assumes the processing load and continues to monitor and manage active calls.
- A load-sharing approach is taken at the CCS and SGU. In the event that one processor fails, the redundant processor assumes the entire signaling load. The failure of a single unit does not result in any calls being dropped.
- Signaling links are equally distributed between the two load-sharing CEs of the CCS and SGU. Full SS7 redundancy is supported, where the operator defines linksets. If a link in a linkset fails, the load is distributed to the remaining active links of the linkset. If an entire card fails, the links associated with the card are bought down and the load is distributed to the remaining active links of the linkset. Signaling links are engineered at a maximum loading factor of 0.4 Erlang. In the event of a failure of up to half of the links, this traffic is temporarily supported by the associated links at no more than 0.8 Erlang.
- An N+1 redundancy scheme is in place for the IP/Interactive Voice Response (IVR). The system is sized to ensure that there is one more IP/IVR than is needed to handle the anticipated processing load for the platform. Because IP/IVRs are exactly equivalent, and because no subscriber-specific data resides at the unit, the failure of one IP/IVR has no effect on system availability.
- Full redundancy is in place at the Platform Manager. Each processing component is replicated.
- The High Speed Backbone Network is configured with two paths between every system component.

Scalability

The Comverse ONE solution has the flexibility to be configured to support widely differing numbers of subscribers. In various sites around the world today, the same platform technology supports as few as 10,000 subscribers and as many as several million subscribers, with virtually unlimited scalability. The platform is designed to grow incrementally without the need to replace the initial hardware and software components. This is achieved by replicating components at the platform as service usage increases. The following table displays each component’s configuration for the support of additional units:

Table 4 Configuration for Support of Additional Units

Component	Configuration for Support of Additional Units
Service Logic Unit (SLU)	N+1 redundancy configuration (IN systems) 2N redundancy configuration (non-IN systems)
Call Control Server (CCS)	2N redundancy configuration
Signaling Gateway Unit (SGU)	2N redundancy configuration
Rating DB	2N redundancy configuration

For further growth, a single the Comverse ONE solution is distributed across multiple sites. Each HSBN is interconnected using data routers. In this case, several sites combine to behave as a single, large system. A distributed system supports many millions of active service subscribers and BHCA. The distributed platform approach is also well suited to a geographically dispersed environment and provides a logical means of expanding the capacity of the system over time. Many configuration options are available. For example, one or more platforms are deployed in different geographic regions, one or more platforms are shared between regions, and so on.

NOTE

The Comverse ONE solution capacity depends on the hardware and software packages of each particular system. For specific capacity analysis information, contact your Comverse support representative.

NOTE

Passwords set at the SLU level must be no more than 10 characters in order to be compatible with the Product Catalog External Interface Parameters password.

Chapter 5

Notifications Overview

Notifications Overview

Notifications are used by the Comverse ONE solution to provide event-based messages to resources external to the solution. The Comverse ONE solution uses the Short Message Service (SMS) and Unstructured Supplementary Services Data (USSD) notifications to notify subscribers about events and account status. Multiple messages are configured, on a per-primary offer basis, to be sent at various times and under different conditions.

When event conditions are met, the Comverse ONE solution builds up notifications messages and tries to send them out through predefined interfaces to the external devices (UPM, HLR devices, USSD). These external devices ultimately deliver the notifications to the subscriber devices. Notifications are generated due to the Comverse ONE solution activities (for example, charging a balance) or customer management activities.

Service Order Interface:

Handset-based notifications are commonly used for:

- Alerting the subscriber that an event (such as balance expiration) is about to occur
- Warning that the account balance is below a configured threshold
- Notice that a periodic charge has been deducted
- Providing charging details for the most recent activity transaction

Message Formats

Messages are formatted as SMS, USSD or TCP/IP.

SMS

SMS notification triggers are defined on a global (system) basis through the PC, and then optionally configured on a primary offer basis with the specific notification text. The following conditions apply to SMS notifications:

- SMS messages are sent to subscribers through the UPM. UPM is capable of communicating with multiple types of SMS systems made by Comverse and other vendors.
- SMS messages are limited to 3000 bytes. Since SMS messages are limited to 160 characters by standard, the external device receiving the message must be able to segment the messages into the required standard size.
- The SMS interface supports both Telnet and TCP/IP connections.

USSD

USSD notification allows primary offer notifications to appear as text messages directly on a handset screen. In a generic SMS notification, the subscriber must open the handset to read the text. USSD notification, in what is normally called a flash, goes through the SGU to the Home Location Register (HLR) gateway, which forwards the notification to the subscriber.

The following limitations apply to USSD notification:

- USSD notification is applicable only in GSM networks.
- Only Standard ETSI GSM Phase 2 and Ericsson Phase 2 are supported.
- All HLRs must support either ETSI GSM Phase 2 or Ericsson Phase 2 in the network.
- USSD message text is limited to 160 characters.
- USSD notifications supports all the provisioned languages not just the default language

HLR

The Comverse ONE solution supports notification through the HLR, the main database of permanent subscriber information for a mobile network. HLR notification is an optional feature that informs the network HLR that certain predefined special events have occurred in the subscriber account. Messages containing updated subscriber data are sent to the HLR and are stored in the HLR database. The HLR sends response and acknowledgement messages to the Comverse ONE solution.

In some cases, the network HLR must be notified about special events that occurred in the subscriber account. For example, with certain subscriber state changes, operators may choose to exclude subscribers from specific activities. By notifying the HLR as soon as the subscriber becomes suspended, it is possible to bar these incoming calls on the HLR.

If the subscriber recharges and reactivates the account, another message is sent to the HLR, re-enabling incoming calls to that subscriber. There are other cases when the HLR needs to be notified. Most are related to account state changes (such as when the subscriber account is activated for the first time).

Currently, HLR messages are limited to 180 bytes.

The Comverse ONE solution supports the following HLR Interfaces.

- **Generic SMS:** For sending SMS messages to the Comverse SMS gateway via TCP/IP.
- **Ericsson HLR:** For sending HLR messages to the Estonia Mediation device via telnet.
- **Siemens HLR:** For sending HLR messages to a Siemens switch via a serial port.
- **Nortel HLR:** For sending HLR messages to a Nortel switch via TCP/IP.

System Level Notification Setup

The system level tasks for setting up notifications are:

- Feature activation
- Setting up triggers
- Setting external interface parameters
- Interface assignment

Notification Triggers

A trigger defines the event and the condition (operator and value) that sets off the notifications.

Each trigger has a name, event, operator, and value. The trigger name is a maximum of 30 characters including special characters. Triggers are set up within the Product Catalog (PC).

Feature Activation

There are instances where a feature must be activated before an event is used in a trigger. However, some trigger events are global and not dependent on features. See Feature-based Notifications on pages 57 and 74.

Notification Generation

Notifications to the subscriber and to the network HLR are generated using the same mechanism. This logic allows defining multiple types of notifications, which are sent under very specific conditions, configurable on a per-primary offer basis. Subscriber (SMS/USSD) and HLR notifications differ primarily in the message destination and that the communication protocol is used.

NOTE For subscribers, there is no single global throttle or shut off tag for notifications. If they are provisioned correctly and the external device is reachable, the notifications are sent. Notifications can be disabled on a specified subscriber account.

Latching

Notifications are checked for latching prior to being sent. It is possible to latch a condition so that once the condition is met and the message is sent (either to the subscriber or to the HLR) the first time, the message is not sent again and again because the condition continues to be matched. If a condition is not latched, the message is sent every time the condition is met.

NOTE It is strongly recommended that operators use latching. Latching effectively minimizes the total number of messages sent.

For example, suppose a condition states Balance <= 50. Obviously, this condition is met during every call after the subscriber crosses the $50.00 barrier. This results in the generation of multiple messages (one on every event), which do not provide any new information to the subscriber.

To prevent this, a notification message condition is defined as latched (Latch On set in primary offer provisioning for the message trigger). Latched conditions generate the message only once when they are met. No additional messages are sent (by this entry) until the condition is found to be false again, or the message condition is unlatched.

In the example, if the subscriber recharges the account with a $100.00 card, the account balance rises above $50.00. The condition is now unlatched (the message is played again) but becomes latched again the next time the subscriber drops below $50.00.

Latching is not applicable in all events. Refer to the following table to determine applicability.

Table 5 Latch Events

Event	Latch?
State	Yes
Balance 1-10	Yes
Balance Expiration Date	Yes
Balance 2-10 Expiration	Yes
Previous State	Yes
Activity_Characteristics	No
Activity_Charge	No
Sub_Notif_Level	No
Balance to Balance	No
Account to Account	No
Recharge, RC, NRC	No
SS_Change	No
Award, NearAward	No

Table 5 Latch Events (Continued)

Event	Latch?
PO Change	No
NRC No Charge, NRC Charge Full	No
Grant	No

Bypass

Bypass has to do with whether the notifications are sent through the SDP notification queues or sent directly to the notification agent.

When notification messages are generated, the system temporarily puts the messages in adatabase queue based on the SDP assignment prior to sending. The queue stores one million messages. The queued messages are sent out to the external device on a first in, first out basis. If the system generates more messages than can be sent out, old messages are overwritten and can be lost. With bypass, the operater can configure the system to not put messages in the queue and instead send them out directly to the external device. If a message has bypass and the external system is not reachable on the first attempt, this messages are not retried.

Going through the queues is more costly in terms of processing requirements, however it is more reliable and ensures that notifications are sent in the order generated. With bypass set, the notifications skip the queue and go directly to the notification agent. This is more efficient but there is no guarantee that the notifications are sent in the order that they were generated.

Also, if the notification agent is down, without queuing, it is possible for a notification to be dropped.

For notifications where order matters (which typically includes HLR notifications) or where it is important to guarantee delivery, bypass should not be used.

In general, this is a technical detail, and probably belongs in the architecture section (where queues are shown).

Notifications are checked for bypass prior to being sent. Bypass is not possible for notifications generated by the Rating DB but is applicable to all URE notifications. The bypass mode must be configured in order for the bypass notification to be generated.

Events

Notification messages are sent to subscribers to inform them that a particular event has occurred with regard to their account. This includes one of the following changes of account state:

- Successful recharge of the subscriber
- A change of subscriber state or balance that causes it to reach a predefined threshold (such as, after a billable call)
- A change of expiration date
- A usage award in the form of a bonus or discount
- A balance transfer, activity characteristic or charge
- Application of a recurring charge
- Completion of a particular usage

Conditions

Notification messages are sent to subscribers to inform them that a particular event has occurred. Each event is associated with a one or more conditions (Operator and Value) that trigger the notifications.

For example, an operator notifies subscribers when their balance falls below a certain amount. In this case, a balance threshold value is defined as the event condition for this notification and the message is triggered when that event condition is first detected. Similarly, when a subscriber earns a usage-based award, the bonus or discount threshold triggers a notification message.

The system provides a set of predefined operators that can be used with each event to set up notifications triggers. This value defines the threshold that actually triggers the notification. Based on the event, the system provides a list of appropriate operators and values.

For feature-dependent events like the near award, the conditions are set up within the feature.

Refer to table 4 which lists the events and conditions (both operators and values) used to define notification triggers within the Comverse ONE solution. The table also indicates whether notifications are generated by the Rating DB, URE or Unified Application Programming Interface (UAPI).

Table 6 Notification Events

Event	Operator	Valid Values	Notes/Reference	Notifications Generated By Rating DB/URE/UAPI			Applies to Subscriber, Account or Both
State	"Equal To" "Not Equal To" "Less Than" “Less Than or Equal To" "Greater Than" "Greater Than or Equal To"	Idle, Active, Await First Recharge, Await activation, Suspended (1-4), Fraud Lockout.	When the state is selected as the event, the predefined subscriber state is used to prepopulate the dropdown list. Any combinations of operator and valid values is used.	Yes	Yes	Yes	Both
Balance	"Equal To" "Not Equal To" "Less Than" “Less Than or Equal To" "Greater Than" "Greater Than or Equal To"	Decimal number. Any numeric value is entered up to 15 digits or ‘.’	Allows setting notifications based on a specific balance within the primary offer. Includes all balances provided by the Comverse ONE solution.	Yes	Yes	Yes	Both
Balance Expiration Date	"Equal To" "Not Equal To" "Less Than" “Less Than or Equal To" "Greater Than" "Greater Than or Equal To"	Numeric	Only whole days are entered 0-9. Up to 8 digits are entered.	Yes	Yes	Yes	Both
Previous State	"Equal To" "Not Equal To" "Less Than" “Less Than or Equal To" "Greater Than" "Greater Than or Equal To"	Idle, Active, Await First Recharge, Await activation, Suspended(1&2), Customer Center Only, Disabled (S4), Fraud Lockout	Allows setting notifications to inform the subscriber about their previous account state.	Yes	Yes	Yes	Subscriber
TML	"Equal To" "Not Equal To" "Less Than" “Less Than or Equal To" "Greater Than" "Greater Than or Equal To"	Numeric	Include the TML value in the account/subscribers TML column.	Yes	Yes	Yes	Account

Table 6 Notification Events (Continued)

Event	Operator	Valid Values	Notes/Reference	Notifications Generated By Rating DB/URE/UAPI			Applies to Subscriber, Account or Both
Activity Characteristic	“Matches Exactly” “Contains One of” “Contains All of”	Any predefined Activity Characteristic	The Activity Characteristic is activated by a system wide parameter- *CHG_NOTIF_ ALLWD* in the *Service_ Parameters* table which must be set to 1 to activate the feature. This is equated to any activity characteristic or group of activity characteristics defined in the system which, when matched for the activity, triggers sending the SMS.	No	Yes	No	Subscriber
Activity Charge	"Equal To" "Not Equal To" "Less Than" “Less Than or Equal To" "Greater Than" "Greater Than or Equal To"	Decimal value (0-9 or ‘.’) up to 15 digits.	The Activity Charge is activated by a system wide parameter- *CHG_NOTIF_ALLWD* in the *Service_Parameters* table, which is set to 1 to activate the feature. This is set to a currency charge value which, when crossed, triggers sending the SMS regarding the activity charges. This is only for the total activity currency charge.	No	Yes	Yes	Both
Sub Notif Level	"Equal To" "Not Equal To" "Less Than" “Less Than or Equal To" "Greater Than" "Greater Than or Equal To"	Number 1-9, up to 3 digits and must be between 1-100.	Activated by a system-wide parameter: CHG_NOTIF_ALLWD in the *Service_Parameters* table, which is set to 1 to activate the feature. This is a subscriber-level notification characteristic to determine if the subscriber is configured to receive the notification. For example, if the subscriber notification level is lower than the trigger level, the notification is suppressed.	Yes	Yes	Yes	Subscriber
AccountNotif Level	"Equal To" "Not Equal To" "Less Than" “Less Than or Equal To" "Greater Than" "Greater Than or Equal To"	Number 1-9, up to 3 digits and must be between 1-100.	Activated by a systemwide parameter: CHG_NOTIF_ ALLWD in the Service_ Parameters table which must be set to 1 to activate the feature. This is an account level notification characteristic to determine if the account is configured to send the notification. If the account notification level is lower than the trigger level, the notification is suppressed.	Yes	Yes	Yes	Account
Balance To Balance	None	None		No	No	Yes	Both
Account to Account	None	None		No	No	Yes	Both
SS_Change	None	None	Previously used for identity change. Renamed to support Shadow Subscriber ID change.			Yes	Subscriber
Award	None	None	Accumulator-based threshold.	Yes	No	No	Both

Table 6 Notification Events (Continued)

Event	Operator	Valid Values	Notes/Reference	Notifications Generated By Rating DB/URE/UAPI			Applies to Subscriber, Account or Both
Near Award	None	None	Accumulator-based threshold.	Yes	No	No	Both
Recharge	None	None		Yes	No	No	Both
Recurring Charge Full	None	None	When defining the RC term, if Notification Required is enabled and if notification text is defined for Recurring Charge Full event in the offer, the subscriber is sent a notification each time the RC is charged in the full amount. A subscriber in post active state is charged if RC Term is configured to be charged in post active state. In this case, the subscriber receives a notification if the full amount can be charged for RC. If no notification text is defined for this event, the subscriber is not sent a notification.	Yes	No	No	Both
Recurring Charge No Charge	None	None	When defining the RC term, if Notification Required is enabled and if notification text is defined for Recurring Charge No Charge event in the offer, the subscriber is sent a notification each time the RC cannot be charged. A subscriber in post active state is charged RC if RC Term is configured to be charged in post active state. In this case, the subscriber receives a notification if the RC cannot be charged. If no notification text is defined for this event, the subscriber is not sent a notification.	Yes	No	No	Both
Recurring Charge Partial Full	None	None	When defining the RC Term, if Notification Required is enabled and if notification text is defined for Recurring Charge Partial Full event in the offer, the subscriber is sent a notification each time the RC is charged in a partial amount. A subscriber in post active state is charged RC if the RC Term is configured to be charged in post active state. In this case, the subscriber receives a notification if the RC is charged partially. If no notification text is defined for this event, the subscriber will not be sent a notification.	Yes	No	No	Both

Table 6 Notification Events (Continued)

Event	Operator	Valid Values	Notes/Reference	Notifications Generated By Rating DB/URE/UAPI			Applies to Subscriber, Account or Both
PO Change	None	None		Yes			Subscriber
NRC No Charge	None	None	When defining the NRC Term, if Send Notification is enabled and if notification text is defined for NRC No Charge event in the offer, the subscriber is sent a notification each time the NRC cannot be charged. If no notification text is defined for this event, the subscriber is not sent a notification.	Yes	No	No	Both
NRC Charge Full	None	None	When defining the NRC Term, if Send Notification is enabled and if notification text is defined for NRC Charge Full event in offer, the subscriber is sent a notification each time the NRC is charged in the full amount. If no notification text is defined for this event, the subscriber is not sent a notification.	Yes	No	No	Both
BalDeferred Grant Activated (Bal 1-N)			Generates notification for both real-time sync and batch processes.	Yes	Yes		Both
BalGrant NearExpire (1-N)	"Equal To" "Not Equal To" "Less Than" “Less Than or Equal To" "Greater Than" "Greater Than or Equal To"	0-99, number of days from the grant expiry date.	Announces when the next grant expire date for that balance is x days away in the same manner as in balance expiry. 0 means today.	Yes	No	No	Account
BalGrant Expired (Bal 1-N)	None	None	Announces a grant expiry for a particular balance.	Yes			Both
PP_ Subscription_ OK	None	None	Indicates the promised payment subscription is okay.	Yes	No	No	Account
PP_ Subscription_ Fail	None	None	Indicates the promised payment subscription has failed.	Yes	No	No	Account
PP_Reversal_ OK	None	None	Indicates a promised payment reversal has occurred.	Yes	No	No	Account
PP_Reversal_ Fail	None	None	Indicates a promised payment reversal has failed	Yes	No	No	Account
PP_Repayment	None	None	Indicates a promised payment repayment has occurred.	Yes	No	No	Account
Partial_PP_ Repayment	None	None	Indicates a promised payment partial repayment has occurred.	Yes	No	No	Account
PP_Due_Date	None	None	Indicates the promised payment due date is reached.	Yes	No	No	Account
Daily_Fee	None	None	Indicates the daily fee has been applied.	Yes	No	No	Account

Table 6 Notification Events (Continued)

Event	Operator	Valid Values	Notes/Reference	Notifications Generated By Rating DB/URE/UAPI			Applies to Subscriber, Account or Both
Subscriber_Add			Notifies when a subscriber is added to an account or transferred to the account.	No	No	Yes	Account
Subscriber_ Disconnect			Notifies when a subscriber is disconnected from an account.	No	No	Yes	Account
Subscriber_ Transfer (out)			Notifies when a subscriber is transferred from an account.	No	No	Yes	Account
Offer/bundle - Disconnect			Notifies when an offer is disabled due to lack of funds to pay a due charge.	No	Yes	No	Both
Offer - Disable			Notifies when offer is disconnected.	No	No	Yes	Both
Offer/bundle - Active			Notifies when an offer/bundle is activated.		Yes	Yes	Both

NOTE Sub_Notif_Level and Activity_Characteristic cannot be primary trigger events. They can only be used in combination with another trigger.

NOTE It is recommended that Activity_Charge only be set in combination with another event to avoid excessive SMSs.

Compound Triggers

The Comverse ONE solution sends notifications based on compound triggers. Compound triggers are based on multiple events where all conditions are true for the notification to be generated. More than two events can be compounded. Compound triggers are based on any valid combination of events. A table of valid trigger combinations is found in Chapter 3.

Compound Trigger Latching

For compound triggers to generate, all conditions must be true. However, latched compound triggers fire again if one of the conditions was false since the initial notification.

Tokens

A token is a placeholder used to represent a dynamic value that is filled in when the notification is sent. For example, the token ~balance~ is replaced by the subscriber's current balance value when the notification is sent.

For example, a message text may be "You have $~balance~ in your account", where the ~balance~ token represents the subscriber balance. When used for a subscriber who has a balance of $20.00, the actual SMS message is "You have $20.00 in your account."

NOTE The available tokens are listed in Chapter 6, Table 16.

Tokens can be used as part of the notification text, which is configured in the primary offer.

More than one token can be inserted into a message. There is no limit to the number of tokens that are inserted into a message. This is limited by the interface message size.

Tokens are usually defined based on feature needs. However, they are all made available for use with all features as long as the feature that introduced the token is on.

When tokens are used in messages, the system does not validate on a feature basis. For example, the system does not validate that trigger events are used with the same tokens that were introduced in the same feature. This puts the burden of provisioning meaningful messages on the operator.

Smart Tokens

Smart Tokens is a mechanism that allows operators to suppress messages if the value of a balance is zero. The following rules apply to Smart Tokens:

- Messages contain Smart Tokens delimited by {" and "}. A sample message is "Hi Dan. {Your balance is ~Balance[2]~./Zero/} Thanks!". The entire smart token is "{Your balance is ~Balance[2]~./Zero/}". The /Zero/ delimiter (or rule indicator) prepares us for the future phase requirements to support other rules (such as /Negative/) and indicates that this is the zero/null suppression rule.
- The entire Smart Token is suppressed if any of the tokens contained within the delimiters { and } is zero or null. For example, if the entire message is "Hi Dan. {Your balance is ~Balance[2]~ and ~Balance[3]~./Zero/} Your balance 4 is ~Balance[4]~. Thanks!" and either Balance[2] or Balance[3] is zero, then the entire Smart Token is suppressed resulting in the subscriber seeing the message: "Hi Dan. Your balance 4 is 432.34. Thanks!"
- If the tokens have values, they are replaced and the delimiters { and } are removed along with the rule /Zero/. Using the above example, if Balance[2] and Balance[3] do have values, then the subscriber sees the message: "Hi Dan. Your balance is 100.45 and 321.22. Your balance 4 is 432.34. Thanks!"
- If suppression of a Smart Token results in an empty notification then that notification is not sent.
- Multiple Smart Tokens occur in a single message but are not nested.
- The PC is not altered. Smart Tokens must be typed in manually by the operator.

Notification Language

In the Product Catalog (PC), under primary offer, the Comverse ONE solution provides a dropdown list of all system supported languages. Once a message is defined, any of the provided languages are then selected. In addition, in the PC, under Subscriber Account, the parameter *Notification language* is used to clarify the subscriber language for notification.

Feature-based Notifications

Different features and sources trigger notifications and it's important to know how they are treated by the Comverse ONE solution. Events and tokens are both dependent on features. If the feature isn't turned on, the token does not apply and the event message is not sent. Activities play an important role as they are grouped for the purpose of activity-based notifications. Definitions must be set in Accounts to determine which parties (roles) are notified regarding balance levels. The Comverse ONE solution also allows account owners to transfer money from their account to another subscriber's account using USSD messaging. Proper provisioning allows account owners to manage their accounts efficiently.

Usage Awards Notifications

The Comverse ONE solution provides two notification event types for the Usage-Based Promotions feature:

- Award
- Near Award

These inform a subscriber that they received an award or are near an award level. This encourages additional usage in order to reach the award. Refer to the *Rating Technical Reference* manual for additional details.

Activity Characteristic Notifications

The user-definable attribute Activity Characteristic allows the network operator to monitor account activities in order to manage activity-based notifications. Up to 16 different characteristics are defined.

For example, this could be used to classify activities in groups of Local, Long-Distance, and International. Another set of classification could be the categories of basic services and premium services.

The the characteristics are not mutually exclusive. Any activity could match multiple characteristics.

The Activity Characteristic is included in the Application Subtype Definition Table. The new Activity_Characteristic notification trigger can be used to configure the generation of a notification based on the Activity_Characteristic.

UAPI

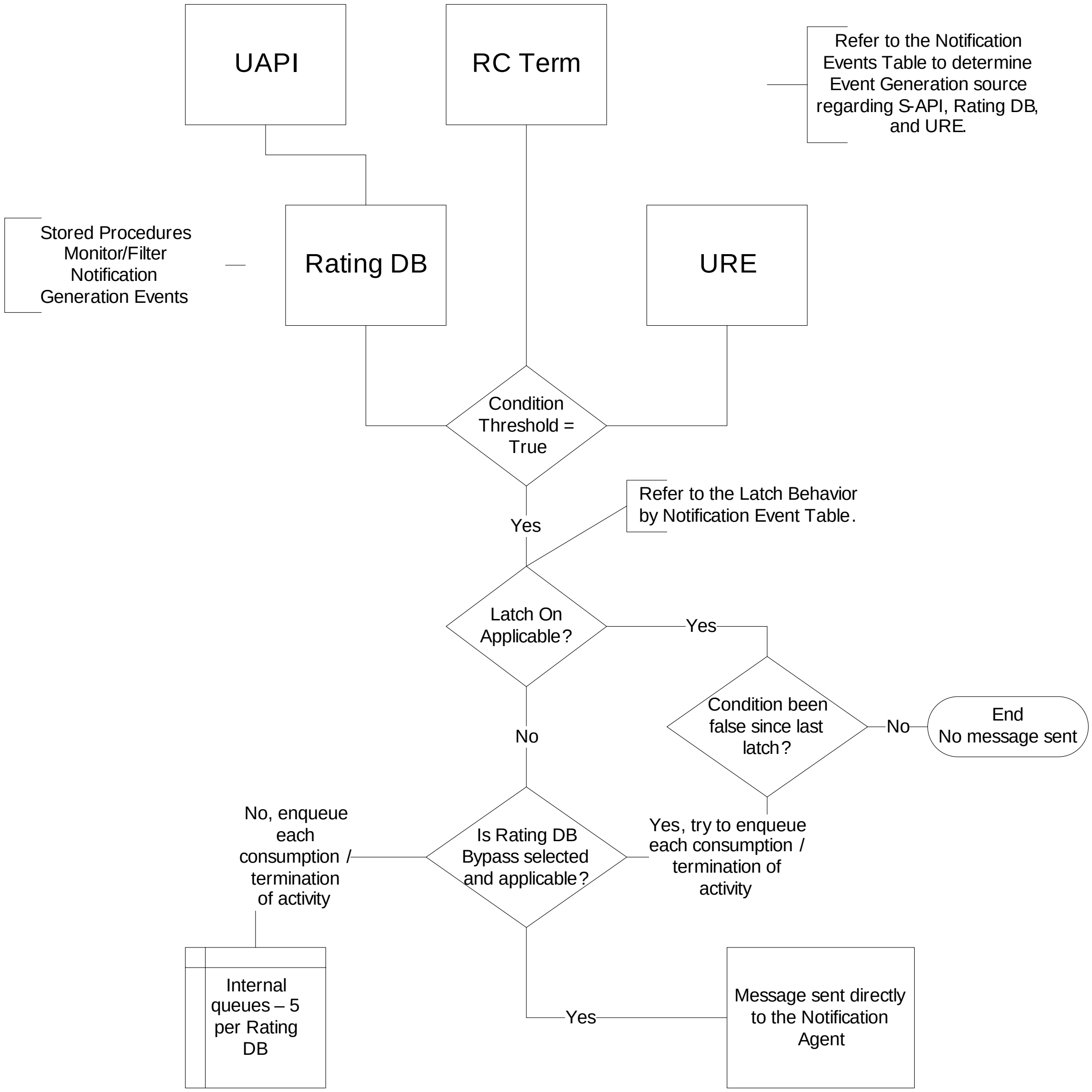
For Customer Center/UAPI activities that result in event (balance, spending limit, and so on) changes, the Comverse ONE solution generates a notification to the subscriber if the event condition crosses a defined threshold.

Notification Processing

Notifications to the subscriber and to the network HLR are both generated using the same mechanism. However, the Comverse ONE solution notifications are triggered by a variety of activities from different sources.

Depending on the notification configuration, activities from CC/UAPI and URE activities trigger notifications. Once the notification activity is detected, the system builds up message queues for each defined interface. Up to five interfaces are supported per Rating DB so there are up to five message queues. This process is referred to as the enqueue process and is depicted in the following figure.

Figure 4 Notification Generation Flow


UAPI
RC Term
Refer to the Notification Events Table to determine Event Generation source regarding S-API, Rating DB, and URE.
Stored Procedures Monitor/Filter Notification Generation Events
Rating DB
URE
Condition Threshold = True
Yes
Refer to the Latch Behavior by Notification Event Table.
Latch On Applicable?
Yes
Condition been false since last latch?
No
End No message sent
No
No, enqueue each consumption / termination of activity
Is Rating DB Bypass selected and applicable?
Yes, try to enqueue each consumption / termination of activity
Internal queues – 5 per Rating DB
Yes
Message sent directly to the Notification Agent

The enqueuing process has the following assumptions and constraints:

- Provisioning is done first:
  - Features are set if any dependent notifications is generated.
  - Triggers are set up.
  - Interfaces are set up and assigned.
  - Within the primary offer, triggers and interfaces are assigned and messages constructed.
  - Account and subscriber notifications are set.
- Message queues are one million messages or more depending on the size of the database.
- Each interface requires a dedicated queue.
- If for any reason the SLU cannot remove (dequeue) messages from the Rating DB queues fast enough, new messages overwrite the oldest messages.

Steps for the above flow are as follows:

1. Stored procedures monitor notification generating activities.
2. For URE activities, the system determines if reservations are required.
    - *a.* If not required, the system attempts to process the notifications at the end of the activity.
    - *b.* If reservations are required, the system attempts to process a notification for each reservation consumption as well as at the end of the activity.
3. Next step checks latch for applicable activities.
    - *a.* If latch is not set for CC/UAPI, the system tries to enqueue the messages.
    - *b.* If latch is set for CC/UAPI, then the system determines if the condition changed to false since the last notification and if so, the system tries to enqueue. If the condition is not false since the last notification, no message is enqueued.
    - *c.* If latch is not set for URE activities, the process continues with a check for Rating DB bypass.
    - *d.* If latch is set for URE activities, then the system determines if the condition has changed to false since the last notification and if so, check for Rating DB bypass (this acts as a new latch). If the condition is not false since the last notification, no message is enqueued.

Notification Handling

In the Comverse ONE solution notification architecture, once notification messages are generated, the system organizes them into interface queues. On each Rating DB, there is one Oracle database queue for each external device. It is also possible to have an instance of the Notification Agent on every SLU.

The Comverse ONE solution maintains a configuration table named Notif_Queue (1-5) which contains:

- An entry for each queue which is identified by the combination of Rating DB and external interface numbers.
- The name of the SLU whose Notification Agent is the primary agent processing messages from the queue.
- The name of the SLU whose Notification Agent is the backup agent processing messages from the queue.

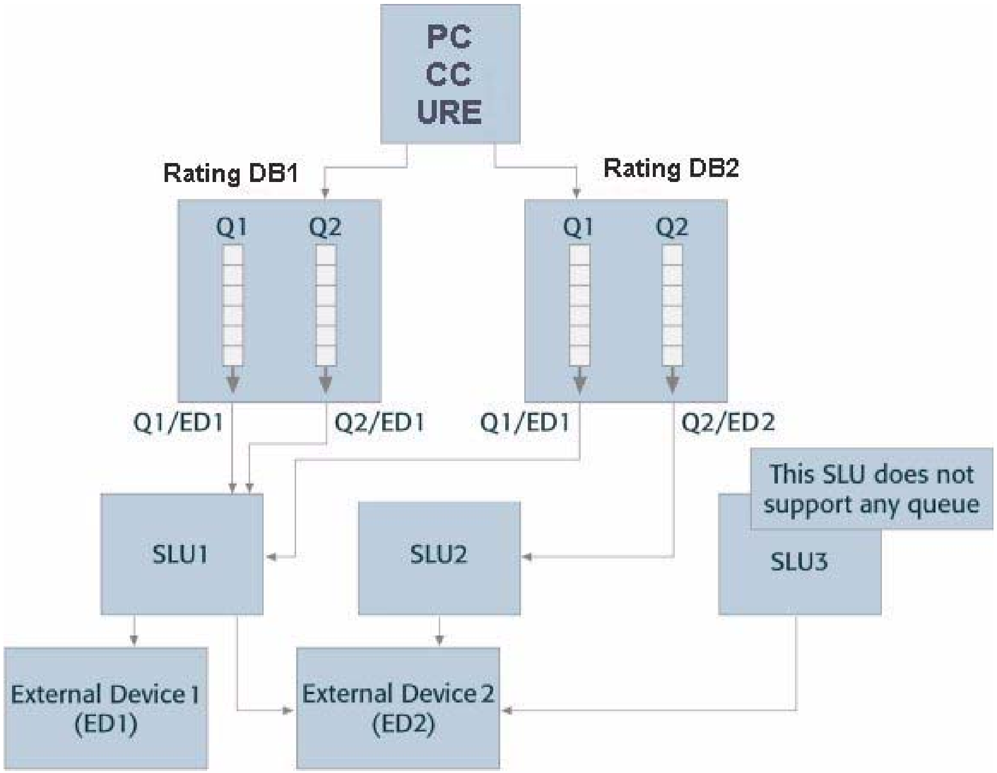
The dequeuing process removes the notifications messages from the Rating DB queues.

To handle database-generated messages, each node has a dedicated thread for each Rating DB attached to it. The number of separate threads on a node equals the number of external interfaces attached to that node plus an additional thread to handle locally generated notifications for each external interface.

High Level Notification Architecture

The following figure shows the high level notification architecture. The Notification Agent and URE processes are within the SLU.

Figure 5 High Level Notification Architecture



This diagram shows the following possible architecture configurations:

1. A system with two Rating DBs, three SLUs, and two external devices. External Device 2 supports multiple connections with one of these connections being separate from all queues and carrying only locally generated notification messages.
2. Additionally, assume the two Rating DBs are separated by a WAN and External Device 1 supports only a single connection (such as a serial HLR interface). Under these conditions, the SLU 1 handles all messages for External Device 1 as well as servicing the queue for External Device 2 on Rating DB 1. In this example, SLU 1 is local to Rating DB 1.

Generic Notifications Interface Specification

The following interface drivers are provided with implied protocols at the Intelligent Peripheral Function (IPF):

- SMS generic via TCP/IP protocol – UPM interface
- hlrEricsson via Telnet
- hlrSiemens via Serial port
- ussdNotification - USSD_IPF, specified at the IPF level

The destination for all messages is specific and configurable in the Comverse ONE solution (specific IP Address and Port). No response is expected.

Usually the third party software/drivers act as servers (such as TCP/IP) to the Comverse ONE solution. They open a socket, listen to it, read from it, and do whatever necessary after receiving the message. They must not send anything back to the Comverse ONE solution, as nothing is expected.

The appropriate message formats and details must be agreed upon between the Comverse ONE solution and the external system vendor. Typically, supported message format includes a 6 byte header, a body of variable length depending on the interface (SMS = 2000, USSD = 160, HLR = 180) and other parameters as provisioned on the Notifications External Interface tab. Some of the parameters are discussed in the following table.

Table 7 Generic Notification Interface Parameters

Field name	Size	Values/Format	Description
Header	6 bytes	Length of message (excluding header) converted to network-byte order using htons– 2-bytes. Fixed bytes of $OK$- 4-bytes. These bytes can be ignored by the external server.	Header information with size of the string and a predetermined prefix $OK$. For USSD flash over SMS, the prefix is usually different.
Body	Variable length	The body contains free text configured on a primary offer basis in the prepaid system. In addition to the free text, tokens are used. Tokens are variable and are filled in with user-specific values. For example, MSISDN (as well as subscriber's current balance and state) are available tokens.	Actual message string
Type		HLR, SMSgeneric, and such	Notification type
SubId		Digit string	Subscriber handset
Direction	2	UP, DN	Threshold crossing direction
Threshold		Digit string	Token threshold value

Example of Generic Notification

When a message text, such as “Hello there”, is sent through a gateway such as UPM SMS, the gateway sees “11$OK$Hello there” (ignoring the quotes) and interprets “11” as the length of the "Hello there" message, not including "$OK$”. The subscriber sees “Hello there” without the quotes.

For example, when sending a notification to the HLR to disable service when a subscriber's balance falls to 10 rubles or below and then re-enable the service when the balance is greater than 10 rubles, the notification messages generated for the subscriber with MSISDN 12345678 are configured as follows:

16$OK$12345678 down 10	Subscriber goes DOWN to 10 Rubles
13$OK$12345678 up 10	Subscriber goes UP to 10 Rubles

Primary Offer Change Notification

Subscribers can receive a notification when there is a change in their Primary Offer. This notification is set up in the Product Catalog.

Refer to the Product Catalog User Guide for details regarding Primary Offer Change notification.

Chapter 6

Subscriber and Account Notification Provisioning

Subscriber Provisioning Overview

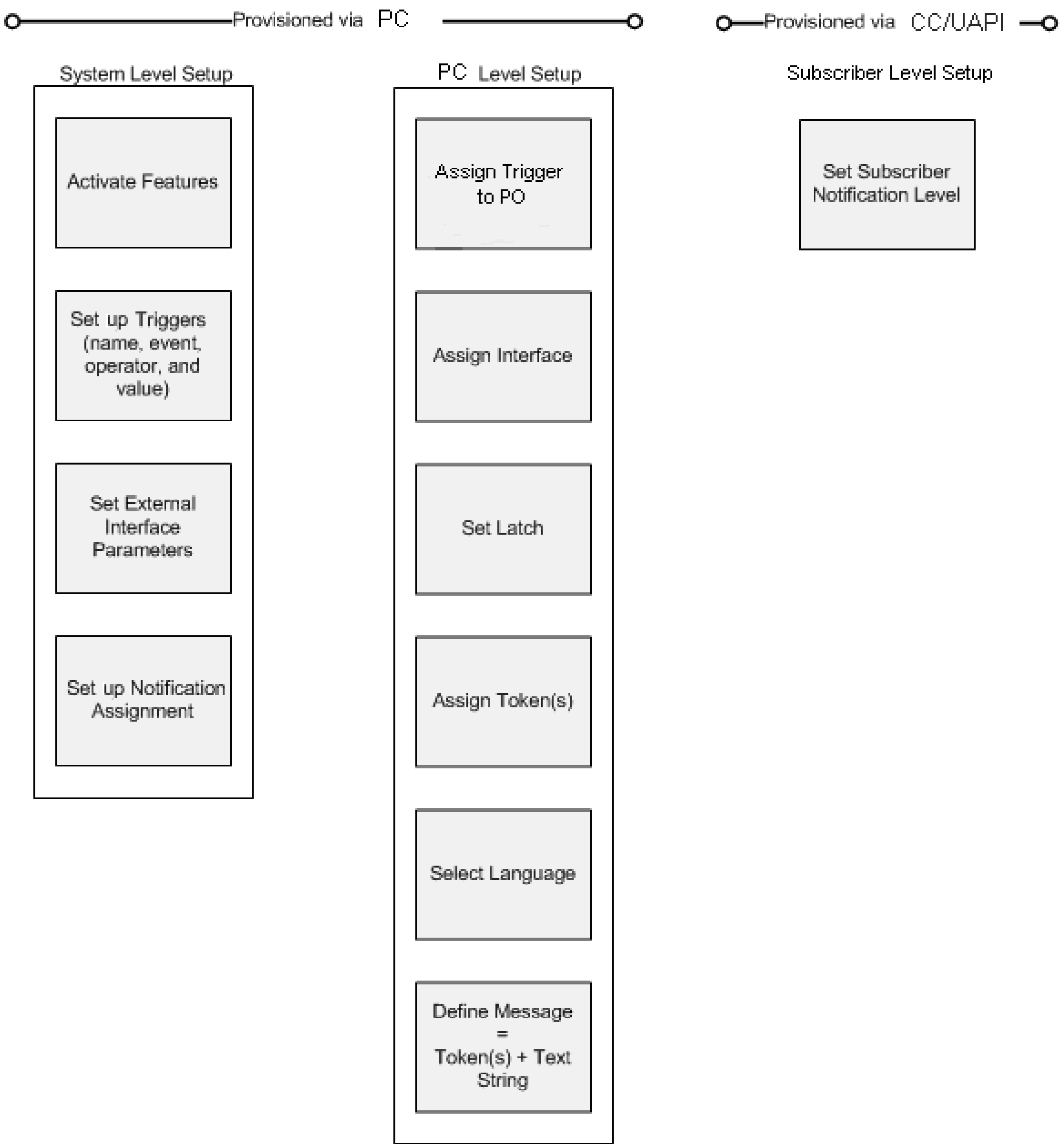
There are three levels of subscriber provisioning for notifications:

- System level
- Primary Offer (subscriber notification) level or Account Notification Template level
- Subscriber level

Refer to the Product Catalog User Guide for specific details regarding provisioning.

The following figure shows the components for provisioning notifications throuch the PC and CC/UAPI.

Figure 6 Notifications Provisioning Components



System Level Notification Setup

The system level tasks for setting up notifications are:

- Feature activation
- Setting up triggers
- Setting external interface parameters
- Interface assignment

NOTE There is no single global throttle or shut off tag for notifications. If they are provisioned correctly and the external device is reachable, the notifications are sent.

Trigger Set Up

A trigger defines the event and the condition (operator and value) that sets off the notifications. There are instances where a feature must be turned on before an event is used in a trigger. Some trigger events are global and not dependent on features.

Each trigger has a name, event, operator, and value. The trigger name is a maximum of 30 characters including special characters.

The Comverse ONE solution allows the following types of notification triggers:

- **One trigger for a single event:** One message is sent to subscribers based on the occurrence of a single event such as low balance, recharge warning, or expiration date.
- **More than one trigger for a single event:** Multiple triggers are defined to generate notifications when different threshold conditions are reached for the same type of event.
- **One trigger for multiple events:** This is known as a compound trigger. For example, a subscriber is notified that their account is Active and they receive a warning that their balance has fallen below $5.00.

Compounding and Validation Rules

Compound trigger combinations are based on the validation rules in the Events section of the Product Catalog.

The following table lists valid compound trigger combinations based on the validation rules.

Table 8 Allowed Compound Trigger Events Based on Validation Rules

Events	Create Stand Alone	Edit	Delete	Compound Trigger
State	Y	Y	Y	Y (all except RC Events)
Balance	Y	Y	Y	Y (all except RC Events)
Subscriber Balance Expiration Date	Y	Y	Y	Y (all except RC Events)
Previous State	Y	Y	Y	Y (all except RC Events)
Recharge	Y	Y	Y	Y (all except Activity Characteristics and RC Events)
Near Award	Y	Y	Y	Y (all except Near Award and RC Events)
Award	Y	Y	Y	Y (all except Award and RC Events)
Spending Limit	Y	Y	Y	Y (all except RC Events)
Activity Characteristics	Y	Y	Y	Y (all except Recharge and RC Events)
Activity Charge	Y	Y	Y	Y (all except RC Events)
Sub Notif Level or Account Notification Template level	N	N	Y	Y (all except RC Events)

Table 8 Allowed Compound Trigger Events Based on Validation Rules (Continued)

Events	Create Stand Alone	Edit	Delete	Compound Trigger
Identity Change	Y	Y	Y	Y (only with Sub Notif Level)
Account to Account	Y	Y	Y	Y (all except RC Events)
Balance to Balance	Y	Y	Y	Y (all except RC Events)
Recurring / Nonrecurring Charge Full	Y	Y	Y	N
Recurring / Nonrecurring No Charge	Y	Y	Y	N
Recurring Charge Partial Full	Y	Y	Y	N
Grant Events	Y	Y	Y	Y (only with Sub Notif Level or Account Notif Level)

NOTE All events can be deleted. All events *except Sub Notification Level and Activity Characteristic* can be edited and created as standalone.

Triggers are set up within the Product Catalog. Refer to the Product Catalog User Guide for details.

Event Trigger Types

It is helpful to establish event trigger types because some tokens are only used with certain types of events. Grouping events into types helps to identify the available tokens for each event.

The following table shows the three major trigger types, associated events, and corresponding tokens.

Table 9 Trigger Types with Associated Events and Tokens

Trigger Type (Global Notifications Screen)	Events	Tokens Allowed on Primary Offer or Account Notification Template
Subscriber	State Balance (1-N) Balance Expiration Date (1-N) Previous State Near Award Award Spending Limit (1-n) Activity Characteristics Activity charge Sub Notif Level Identity Change Account To Account Balance To Balance	Subscriber SMS tokens Service Parameter-controlled tokens ALCO tokens NextGen tokens
Voucher	Recharge	Subscriber SMS tokens Voubher tokens Service Parameter-controlled tokens ALCO tokens NextGen tokens
Recurring charge	Recurring Charge Full Recurring Charge No Charge Recurring Charge Partial Full	Subscriber SMS tokens RC tokens Service Parameter-controlled tokens ALCO tokens NextGen tokens
Promised Payment	PP_SUBSCRIPTION_OK PP_SUBSCRIPTION_FAIL PP_REVERSAL_OK PP_REVERSAL_FAIL PP_REPAYMENT PARTIAL_PP_REPAY PP_DUE_DATE	Promised Payment tokens can only be used with these events.
Grants		
Daily Fee	Daily_Fee	Only Daily Fee tokens and Subscriber SMS tokens can be used with this event.

Notification Interfaces

The interface provisioning process defines the primary and secondary communication parameters for the external devices. The external interfaces and communications parameters allow external interface devices to receive the notification message information.

NOTE All modifications to parameters must be made in a future version. The current version cannot be modified.

When defining External Interfaces, the parameters listed in the following table can be provided for each interface.

Table 10 External Interface Parameters

Parameter	Data Type / Size	Valid Values	Notes	Required / Optional
Name	30 characters - text plus special characters	Each interface name must be unique system-wide.		Required
Driver	Dropdown menu - 5 entries are provided.	The following drivers (and an implied protocol for each) are provided at the Notification Agent: smsGeneric tcp/ip hlrEricsson ---> telnet hlrSiemens ---> serial port ussdNotification ---> USSD_IPF Drivers cannot be mixed and matched. Even though drivers are set on PC and not validated (except for USSD), protocols are implied by the driver name.	This dropdown menu is populated from the entries in the *Notif_Driver* Table When ussdNotification is selected, all remaining fields on the External Interface Parameters Detail panel are unavailable.	Required
Protocol	Selection – Dropdown	serial tcp/ip telnet	Values are taken from the *NOTIF_PROTOCOL* table.	Required
Primary IP Address	Alpha-numeric	Any IP address Value is not validated on entry.	Specifies the IP address of the external device.	Required or system uses default of 0.0.0.0.
Primary Port	Numeric; 5 digits	Any port number Value is not validated on entry.	Destination host port number	Optional
Retry Count	Numeric; 5 digits	Values 0-9	Specifies how many times the system tries to resend the message in case of errors.	Required
Retry Interval	Time in seconds, Numeric - 5 digits.	Values 0-9	Specifies the time interval before retry.	Required
Protocol Prefix	Text- 100 characters	Any combination	The Protocol prefix must be limited to 100 characters and is validated according to the rules of the string as defined by specific interface protocol.	Optional, default blank
User Name	Text 30 Characters	Any combination	For those interfaces that require a username.	Optional, default blank
Password	10 characters	Any combination of characters	The password is encrypted on entry.	Optional, default blank
Secondary IP Address	Entry box- IP format	Any valid IP address	Specifies an alternate external device address.	Optional, default blank

Table 10 External Interface Parameters (Continued)

Parameter	Data Type / Size	Valid Values	Notes	Required / Optional
Secondary Port	Numeric – 5 digits	Numeric – 5 digits	Any valid port number	Optional, default blank
Interface Supports Multiple Connection	Check Box		Specifies if the device accepts multiple simultaneous connections. For example, the serial HLR device supports only one connection.	Optional, default unchecked
Allow SLU to Bypass Rating DB	Check Box	This check box is only activated if the “The Interface Supports Multiple Connection” is checked.	If this is checked (set to Y), the SLU-generated notifications messages are sent to the local notification agent instead of the queue.	Optional, default disabled

SMS Notifications

SMS notification triggers are defined on a global (system) basis through the PC and then optionally configured on a primary offer or account notification template with the specific notification text.

For the activity characteristic-based SMS notifications, a system-wide flag *CHG_NOTIF_ALLWD* in the *Service_Parameters* table controls the availability of the three user-definable trigger events. If the flag is set to No (default value), the triggers are not available via the PC and does not allow the new notification triggers to be provisioned.

- SMS messages are sent to subscribers through the UPM. UPM is capable of communicating with multiple types of SMS systems made by Comverse and other vendors.
- SMS messages are limited to 2000 bytes.
- The SMS interface supports both Telnet and TCP/IP connections.
- SMS messages are also targeted to an HLR. In this case, specific HLR interface conditions apply.

NOTE

Flash SMS is a type of SMS where the text message appears directly on the handset screen, eliminating the need for the subscriber to open the text message. For Flash SMS notification to work, the SMS Gateway (NLP) and the SMSC must support the interface and the capability.

NOTE

For information about SMS interface details, format language, and format protocol, contact the Comverse ONE solution Tier 4 Support.

USSD Notifications

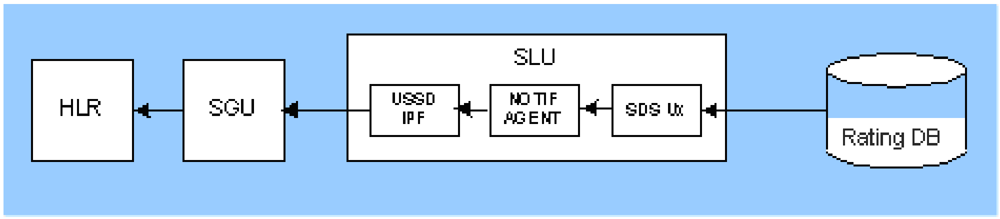
USSD notification allows notifications to appear as text messages directly on a handset screen. USSD notification supports all the provisioned languages, not just the default language.

The following limitations apply to USSD Notification:

- USSD notification is applicable only in GSM networks.
- Only Standard ETSI GSM Phase 2 and Ericsson Phase 2 are supported.
- All HLRs must support either ETSI GSM Phase 2 or Ericsson Phase 2 in the network.
- USSD message text is limited to 160 characters.

The following diagram shows a high level view of the USSD notification architecture.

Figure 7 USSD Notification Architecture



The USSD Notification is passed from the Rating DB to the NOTIF AGENT process via the SDS Ux. This is identical to the way all other notifications are passed to the NOTIF AGENT process. The NOTIF AGENT process then sends the notification to the USSD IPF process which sends it to the HLR via the SGU. The HLR in turn sends the notification to the mobile subscriber (MS) via the MSC. The HLR passes indication of failure or success back to the USSD IPF process via the SGU.

The following assumptions apply to USSD notifications:

- No response is sent from the USSD IPF process to the NOTIF AGENT process indicating the status of delivery.
- USSD messages are not stored anywhere in the network and are dropped if the MS is inaccessible.
- There is no retry mechanism within the SLU in case of errors. The only indication of error is in the USSD IPF measurements.

NOTE Retry is the attempt to *reconnect* to the external interface, not to *resend* a message to the external interface.

- USSD Notifications are sent over the standard SMS interfaces, however, a special code specified by UPM (and transparent to the Comverse ONE solution) must be defined in the payload (for example, OF instead of OK).

The following information is specific to USSD notifications:

- USSD notifications go directly to the handset display. If multiple USSD notifications are sent, each new message overwrites the previous one.
- USSD notifications give the subscriber a similar end-user experience (no need to open the notification) and have the added benefit of not using SMSC resources.
- One drawback of USSD notifications is that it is limited to GSM networks.

Notifications Through HLR

The HLR stores and updates a variety of subscriber account information used to determine call handling. In some cases, it is important to notify the network HLR about special events that occurred in the subscriber account. The Comverse ONE solution supports four HLR interfaces which are listed in the following section.

The interface is defined and then assigned to an SLU.

HLR Interfaces

When subscriber data changes, messages containing updated information are sent to the HLR and stored in the database. The HLR sends both response and acknowledgement messages to the Comverse ONE solution.

NOTE There is no standard interface to send such messages to the network HLR. Unfortunately, every brand of HLR has a different interface and uses different commands to achieve similar results. This requires switch vendor-specific drivers to be implemented on the Comverse ONE solution with each driver adapted for a specific HLR interface. If the subscriber profile changes, this usually requires the use of Man Machine Language (MML) interfaces to the HLR.

Currently, the Comverse ONE solution supports the following HLR Interfaces.

- **Generic SMS:** For sending SMS messages to the Comverse SMS gateway via TCP/IP.
- **Ericsson HLR:** For sending HLR messages to the Estonia Mediation device via telnet.
- **Siemens HLR:** For sending HLR messages to a Siemens switch via a serial port.
- **Nortel HLR:** For sending HLR messages to a Nortel switch via TCP/IP.

HLR messages are limited to 180 bytes.

Notification (Interface) Assignment Provisioning

Once the interfaces are defined, the next step is to assign each interface to an SLU. Notification assignment is the process of identifying which SLU (Notification Agent) services a specific interface queue on each Rating DB. This is the SLU that forwards the message to the external device.

A Comverse ONE solution configuration has the option to include multiple Rating DBs and multiple SLUs. For each Rating DB, the system allows provisioning of a queue for each defined interface with a maximum of 5 queues per Rating DB.

The system allows and requires that each defined interface be assigned specific SLUs (Notification Agent) that is responsible for processing the notifications message in the queue.

The assignment allows for primary and secondary SLU assignments. An SLU can be dedicated to just one external device queue that is consolidated from multiple Rating DBs.

To define an association through the Notification Assignment tab, the following parameters are provided.

Table 11 Notification Assignment Data

Parameter Name	Data Type/Size	Valid Values	Description	Required / Optional
External Interface	Dropdown list	Any interface value defined on the external interface parameters.	Currently only five interfaces are set up per the Comverse ONE solution.	Required
Rating DB	Dropdown list	A valid Rating DB from the dropdown list. The actual process of defining Rating DBs is a deployment issue.	Defines the Rating DB Queue.	Required
Primary Notification Agent	Text (no limit)	Any Name as defined in the Notification agent deployment	Indicates the primary SLU (notification agent) responsible for dequeuing the notifications message and forwarding to the external interface.	Required
Secondary Notification Agent	Text (no limit)	Any Name as defined in the Notification agent deployment	Indicates the back up SLU (notification agent) responsible for dequeuing the notifications message and forwarding to the external interface.	Required

Feature-based Notifications

The following section covers the provisioning of feature-based notifications and how they are treated by the Comverse ONE solution. Events and tokens are both dependent on features. If the feature isn't turned on, the token does not apply and the event message is not sent. The following table shows events and the features that must be turned on for the event message to be sent. This table is used along with the Tokens table.

Table 12 Notification Assignment Data

Event	Feature
Activity Characteristics	Activity Characteristics
Activity Charge	Activity Characteristic
Account to Account	Account
Balance to Balance	Account
PP_ENABLE_FLAG	Promised Payment triggers
DAILY_FEE_ENABLE_FLAG	Daily_Fee trigger

Usage Awards Notifications

The Comverse ONE solution provides two notification event types for the usage-based promotions feature:

- Award
- Near Award

These inform a subscriber that they have an award or are near an award level, thereby encouraging the additional usage required to reach the award.

The award-based notifications are controlled by a global parameter NEAR_AWARDS_ENABLED which enables or disables PC access to the notification type.

This parameter has the following functions:

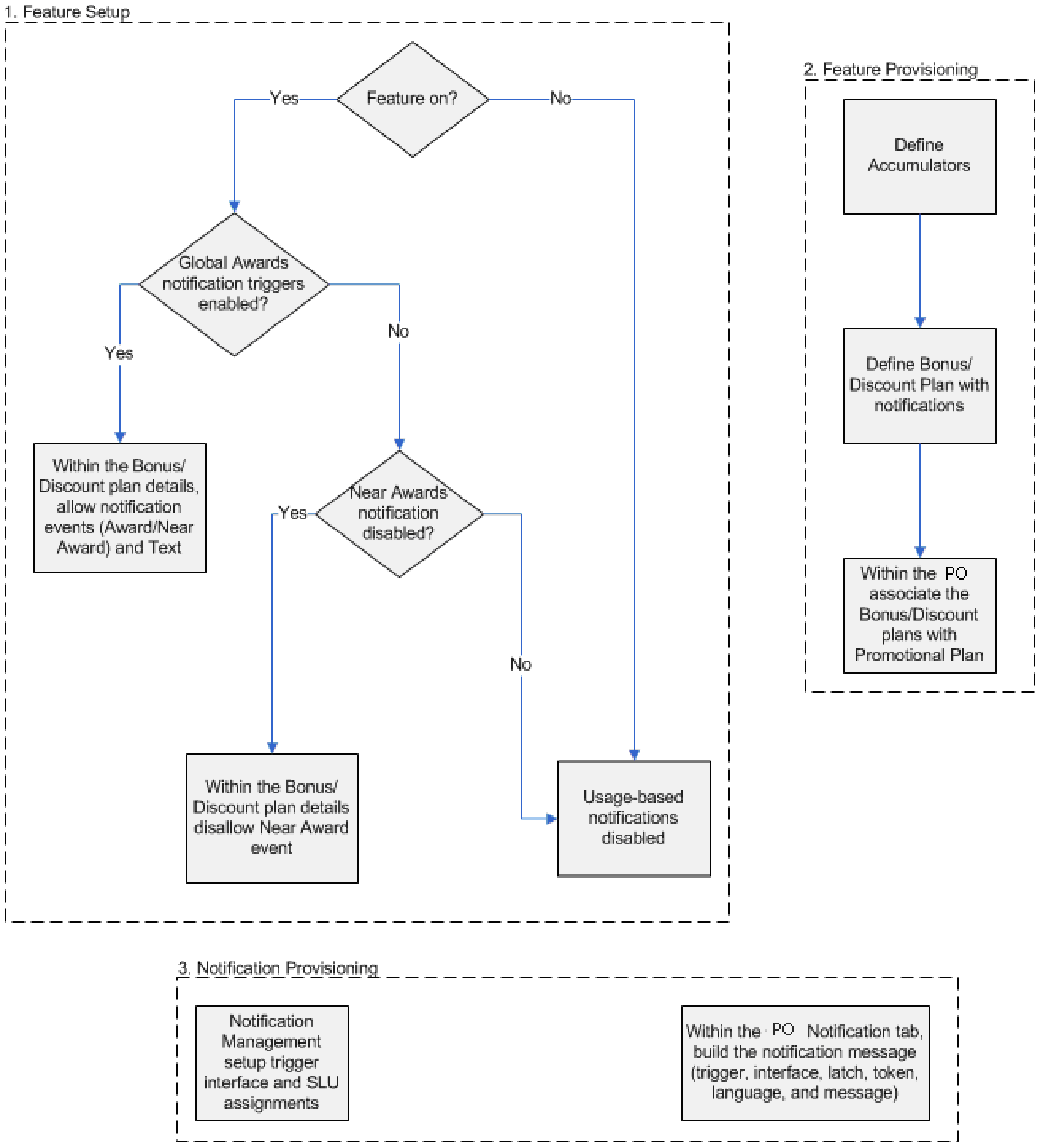
- Turn on or off usage-based notifications entirely
- Disable Near Award notifications but allow Award notifications

This parameter is set when the Comverse ONE solution is installed. The default value is disabled.

NOTE Usage-based promotions events are feature-controlled only at the Bonus/Discount Plan level – not at the Global and SMS Notifications level.

When defining award notifications, trigger conditions are based on the value of the particular accumulator threshold associated with the usage award.

Figure 8 Award-Based Notification Flow


1. Feature Setup
Feature on?
Yes
No
Global Awards notification triggers enabled?
Yes
No
Within the Bonus/ Discount plan details, allow notification events (Award/Near Award) and Text
Near Awards notification disabled?
Yes
No
Within the Bonus/ Discount plan details disallow Near Award event
Usage-based notifications disabled
2. Feature Provisioning
Define Accumulators
Define Bonus/ Discount Plan with notifications
Within the PO associate the Bonus/Discount plans with Promotional Plan
3. Notification Provisioning
Notification Management setup trigger interface and SLU assignments
Within the PO Notification tab, build the notification message (trigger, interface, latch, token, language, and message)

The Award-based notification provisioning process starts with activating the feature. If the Usage Award feature is on, and the accumulators and the award based notifications are on, then bonus and discount plan notifications are provisioned using the PC.

To build a meaningful notification message, the `RewardNotifText` token specific to awards is provided in the global token list. This token allows the notification text defined within the award (Bonus/Discount) to be included in the actual notification message.

The actual notification text is divided into an award-specific message and a primary offer-specific message. The RewardNotifText is limited to 160 characters.

Within the primary offer, the network operator configures the notification text the same as for any other notification. The operator configures a basic text message for Near Awards and a different basic text message for Awards. However, the operator includes the `RewardNotifText` as part of the text message to give specific details about the Award or Near Award.

The actual notification triggers are based on the value of the particular accumulator.

The notification type and award-specific text `award_token` are configured as part of the Global Award (Discount or Bonus) definition.

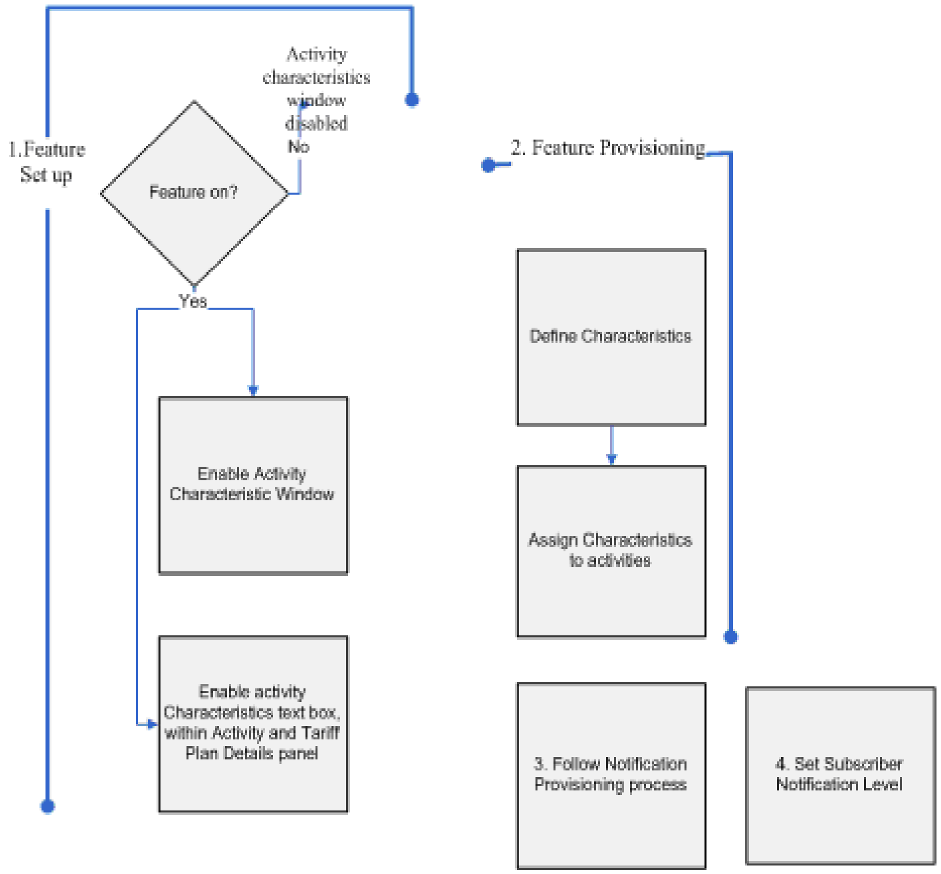
The primary offer-specific handling is configured as part of the primary offer notification configuration.

Activity Characteristic Notifications

The user-definable attribute Activity Characteristic allows the network operator to manage account activities for the purpose of activity-based notifications. Up to 16 different characteristics are defined. The Activity Characteristics act as a set of flags representing different activities.

The following flow diagram shows the Activity Characteristic-based notifications provisioning process.

Figure 9 Activity Characteristic Notification Process


1.Feature Set up
Feature on?
Activity characteristics window disabled
No
Yes
Enable Activity Characteristic Window
Enable activity Characteristics text box, within Activity and Tariff Plan Details panel
2. Feature Provisioning
Define Characteristics
Assign Characteristics to activities
3. Follow Notification Provisioning process
4. Set Subscriber Notification Level

The Activity Characteristics-based notifications provisioning process starts with the feature activation. Once the feature is activated, use the PC to complete configuration. Up to 16 Activity Characteristics are defined.

The following apply to Activity Characteristics:

- The detail pane is used to define a new Characteristic.
- Each Activity Characteristic has an ID and Name. The ID numbers, 1 to 16, are system-generated with a default name Unassigned. The ID numbers cannot be modified.
- The Activity Characteristic Name is up to 38 characters long. A name must be assigned. The only naming restrictions are that the name cannot contain only space characters, and the name Unassigned, which is the default indicating that this Activity is not provisioned yet, cannot be used.
- Activity Characteristics are used to classify activities as Local, Long Distance, International, and so on. Another example of a classification set is basic services and premium services.
- Characteristics are not mutually exclusive. Any activity is assigned multiple Activity Characteristics.
- The Activity Characteristics are included in the Application Subtype Definition Table.
- The new Activity Characteristic notification event is used to configure a trigger to generate a notification.
- No new user roles are required. Any existing roles that view or modify the Application Subtype Definition table are able to create Activity Characteristics.

- The Name column is updateable. The user has the option to provision all of the Activity Characteristics. Only the provisioned activities are stored in the database.

Once the Characteristics are defined, the next step is to associate the Characteristics to Activities.

The Comverse ONE solution supports many-to-many relationships between Activities and Characteristics. Characteristics defined in the previous step appear in the details panel together with other available Activities. From here they are assigned to any desired Activity. More than one Characteristic is assigned to an Activity.

Users select any available number of activities and assign them to the Activity and Tariff Plan.

The next step is setting the notification event triggers.

The following three user-definable events are used to define Characteristic-based triggers:

- **Activity_Characteristic trigger:** Equated to any Activity Characteristic or group of Activity Characteristics defined in the system which, when matched for the Activity, triggers sending the SMS.
- **Activity_Charge trigger:** Set to a currency charge value which, when crossed, triggers sending the SMS informing the Activity charges.
- **Sub_Notif_Level trigger:** Looks at a subscriber-level notification characteristic to determine if the subscriber is configured to send the notification. The actual subscriber notification level is set up through the CC.

The Sub_Notif_Level trigger cannot be used by itself to generate a notification. It is only used in conjunction with other triggers as a qualifier. For example, the system only sends a defined notification if the subscriber’s notification level is set to (or above/below) a specific value.

When creating a new trigger, the Sub_Notif_level event is disabled. However, the Sub_Notif_level event is allowed as an addition to an existing trigger.

With this feature, three operators are available to compare the current Activity Characteristic to the provisioned characteristic for notifications.

Examples of Activity Character Notifications

Suppose that the complete set of Activity Characteristics that is provisioned includes long distance calls, local calls, calls on Sunday, calls on weekdays, Basic Service, and Premium Service.

Suppose further that only the long distance calls and the calls on Sunday Activity Characteristics are provisioned via PC:

Provisioned AC = "Notify me when I make long distance call on Sunday."

Current AC = "Long distance call is made on Sunday with Basic Service."

This example provides a basis for understanding the operators:

- Matches exactly
- Contains all of
- Contains one of

Matches Exactly: This operator checks whether the Provisioned AC completely overlaps the Current AC. For example, the two don’t match exactly because the current AC has one extra characteristic (Basic Service) as compared to the Provisioned AC.

Contains All of: This operator checks whether the current Activity Characteristics satisfy the Provisioned Activity Characteristics. The Current AC must have all those characteristics set that are required by the

Provisioned AC. For example, this includes the long distance calls and the calls on Sunday characteristics. It does not matter if the Current AC has additional characteristics set. Additional characteristics are ignored.

The Current AC satisfies all the Provisioned AC, so the notification is sent.

Contains one of: This operator checks whether at least a single Activity Characteristic is common in Current AC and Provisioned AC**.** When a single characteristic is matched, the notifications is sent.

The final step in Activity Characteristic notification is to set up the subscriber level within the CC Subscriber Details panel. The subscriber level is also set up through UAPI.

Primary Offer and Account Notification Template Provisioning

The next step in notification setup is the association of the Trigger and the Interface to Primary Offer and Account Notification Template. This is done through the PC.

This allows the configuration of the following types of notifications for subscribers in the primary offer and account notification template:

- **Single notification trigger:** A message is sent to subscribers based on the occurrence of a single event, such as low balance, recharge warning, or expiration date.
- **Two notification triggers for the same type of event:** Allows the Comverse ONE solution to notify a subscriber when different threshold conditions are reached for the same event.
- **Single notification trigger based on multiple events:** This is also known as a compound trigger. For example, notifying a subscriber that they are Active and sending a warning message when their balance is below $5.00.

For example, subscribers receive an early warning message that is latched (Latch On) so that a notification message plays only once when their balance has reached or gone below a threshold of $50. They then receive a second notification that is not latched (Latch Off). This becomes a constant reminder when their balance has reached or gone below a threshold of $20, and they continue to receive this notification message until their account is recharged.

Only the subscribers within the primary offer or account notification template get the assigned notification.

The following table lists the Product Catalog Primary Offer Notification Tab parameters with details regarding descriptions and whether the parameter is required or optional.

Table 13 PC Primary Offer Notification Tab Parameters

Parameter	Data Type / Size	Valid Values	Description	Required/Optional
Notification	Dropdown	Pre-populates with values from the notification triggers list.	Attaches a trigger (event) with a primary offer.	Required
Interface	Dropdown	Must have been set up on the notifications Tab.	Specifies the message destination, Currently only 5 interfaces are set up per the Comverse ONE solution.	Required
Latch	Dropdown	Latch Off Latch On	Indicates if the notification is latched if condition is met multiple times.	Optional with a default of No.

Table 13 PC Primary Offer Notification Tab Parameters (Continued)

Token selection	Dropdown	See table for all Token listing	Provides a list of available tokens	Optional. If none selected, token is inserted in the message.
Message	Free text format - size is limited by the interface.	Free format text	Message = Message + Token + language This is the actual message that the subscriber sees. The token is replaced with the actual data base value for the event.	Required
Language	Dropdown	Must be among the provisioned systems language. Hebrew is also supported.	Notifications are in this language regardless of the system and primary offer/account notification template language.	Required

Notifications/Service Order Interface

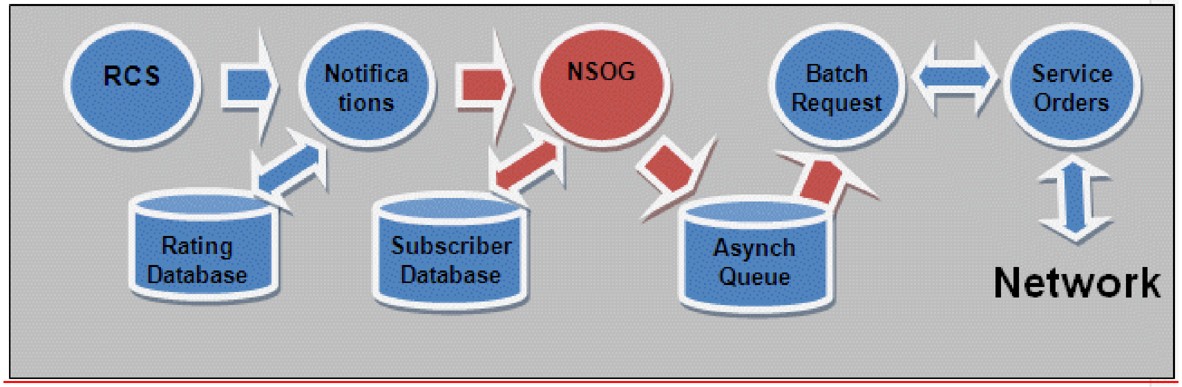
A notifications interface is used to capture service order requests generated by current notifications events. Called the Notifications/Service Order Gateway (NSOG), this mechanism processes certain messages which result in the generation of a service order.

NSOG provides the following configuration options within the Product Catalog:

- Ability to configure the new interface.
- Ability to configure events/triggers.
- Ability to assign notification triggers to the new interface and to generate required service orders.
    - Ability to latch.
    - Ability to configure service order messages including action to be taken based on the event and parameter to identify affected service.
- Ability to generate notification assigned to the new interface and queue them appropriately.
- Ability to trigger service order requests and initiate the requested network provisioning action based on information received from the new interface.

The following figure illustrates the notifications/service order interface flow.

Figure 10 Notifications Service Order Interface



The notifications/service order interface uses the following procedure:

1. The Recurring/Nonrecurring Charge Server (RCS) initiates the process by triggering a Notification of Insufficient Balance using the existing RCS interface to the Notifications module.
2. The Notifications module handles the notification based on configuration and accesses the Rating DB as needed.
3. The NSOG receives the Notification of Insufficient Balance and constructs an XML record containing the information required by the Service Orders module to perform a Subscriber Change order to disconnect the service. NSOG accesses the Subscriber Database as needed to obtain this information. For example, NSOG obtains the target Service ID either from the notification or by querying the database.
4. NSOG places the XML record it created onto an asynchronous queue.
5. The Batch Request module responds to the queued requests and initiates a Subscriber Change order to disconnect the service via Unified API.
6. The Service Order is handled normally and status information is returned to the Batch Request module.

NOTE Notification queues are circular, allowing old notifications to be overwritten by new ones. The above process may cause old notifications to be overwritten if there’s an unusual delay between creation (step 1) and reading (step 3).

Service Order and Network Provisioning

The following provide steps detailing the existing capability for ordering and network provisioning. An example follows the steps.

1. The PC data model has entities called provisioning items and service items. An operator defines them to correspond to the realities/service/resources in the network and then associates them into offers.
2. Each provisioning item and service item may be tied to an extended atrribute. When an offer is instantiated on a subscriber, the related extended data is autopopulated at the offer level or subscriber level.
3. When the subscriber’s extended data is updated by UAPI, an order is generated and a workflow is optionally generated. The workflow may be configured for ordering fulfillment using the existing workflow engine and network provisioning utility.

The difference between a provisioning item and a service item is that a provisioning item is associated with initial AUT in the PC data model. This allows it to actually map to the network provisioning necessary to enable the given collection of usage activities, such as International Roaming service. Service item is a nonusage service, such as voice mail service or Ringback tone.

Example 1:

The operator defines International Roaming Service (IRS) as a provisioning item associated with a Mobile Primary Offer. It is tied to an extended data attribute IRS and this attribute is managed at the subscriber level. At the tim of new customer acquisition, when this offer is assigned to the subscriber, the IRS is set to FALSE by default.

If later, the Subscriber Update UAPI method is called to update the IRS to TRUE, a Service Update service order is created and a workflow may be triggered to enable the international roaming service.

Example 2:

MT SMS is a service item associated with a Mobile Primary Offer. It is tied to an extended data attribute MT_SMS. At the time of new customer acquisition, when this offer is assigned to the subscriber, MT_SMS is set to TRUE by default. If later, the Subscriber Update UAPI method is called to update MS_SMS to FALSE, a Service Update service order is created and a workflow may be generated to disable MT_SMS in HLR.

Message Parameters

The actual message sent is defined within the primary offer, along with any necessary conditions. The message contains any text and has the option to include subscriber-specific information such as the subscriber's current balance or expiration date. These fields are stored as predefined tokens and are replaced by the actual subscriber data when the message is sent.

For example, a message text is "You have $~balance~ in your account", where the ~balance~ token represents the subscriber balance. When used for a subscriber with a balance of $20.00, the actual SMS message is "You have $20.00 in your account." Other available tokens include the subscriber ID, the expiration date, the IMSI, the account state, and a bonus or discount award.

In addition to the message text, the message destination is also specified. This is the SMS Gateway if the message is sent to the subscriber or to an HLR protocol driver if the message is targeted toward the HLR.

HLR messages contain different text than SMS messages. They contain the MML command needed for a given HLR model and make. The message tokens mentioned above, such as the subscriber ID, are used in the messages to let the HLR know which subscriber is affected by the command (such as barred or enabled).

Latching

It is possible to latch a condition so that once the condition is met and the message is sent (either to the subscriber or to the HLR) the first time, the message is not sent again because the condition continues to be matched. If a condition is not latched, the message is sent every time the condition is met.

For example, if a condition states "Balance <= 50", this condition is met during every call after the subscriber crosses below the $50.00 barrier. That results in multiple instances of the same message generated for every event without any new information provided to the subscriber.

To prevent this, a notification message condition is defined as latched (Latch On set in primary offer service provisioning for the message trigger).

NOTE With latch off, all applicable notifications are sent to the subscriber upon any change to the subscriber record. There is the potential of multiple notifications being sent to the subscriber.

For example, if a subscriber has an SMS balance below a certain threshold and makes a voice call (thus affecting the subscriber record), the notification regarding the SMS balance is sent out. This notification is sent every time the subscriber record is touched until SMS balance is recharged. This occurs only if latch off is set on this balance.

Latched conditions generate the message only once when they are met. No additional messages are sent (by this entry) until the condition is found to be false again or if the message condition is unlatched.

In the example above, if the subscriber recharges the account with a $100.00 card, the account balance rises above $50.00. The condition is now unlatched (the message is played again) but becomes latched again the next time the subscriber drops below $50.00.

Notifications have the option of being configured to generate at the end of each activity or to be latched.

Activities from CC/UAPI follow the latch configuration.

For URE-generated activities, notifications have the option of being generated at the end of each activity or being latched.

- For those activities requiring no reservations, the notification enqueuing is done at termination (such as recharge).
- For those activities that require reservations, notification are enqueued at each consumption and termination.

Activities from CC/UAPI consider the latch configuration. The following table shows the effect of latch on notification events.

Table 14 Latch Behavior by Notification Event

Event	Latch Applicable	Example	Latch Off	Latch On
State	Yes	State = Active Activity = Making a voice call.	Notifications are generated for each voice call. This notification is also generated for all other attempted activities.	No notification is sent until the state changes from Active and then back to Active.
Balance	Yes	Balance threshold crossed Balance threshold <=$50 actual balance = $30	Notification generated for all activities.	No notification is sent until the balance goes above $50 and then to <=$50.
Balance Expiration Date	Yes	Expiration threshold <= 10 days	Notification generated for all activities.	No notification.

Table 14 Latch Behavior by Notification Event (Continued)

Event	Latch Applicable	Example	Latch Off	Latch On
Previous State	Yes	Previous_State = Suspended Activity: Make a voice call	Notifications are generated for all voice calls, as long as the previous state is suspended	No notification after the first one. This is maintained until the previous state is changed such that the condition is false and then changed again so that the condition is true.
Activity_ Characteristics	No			
Activity_Charge	No			
Sub_Notif_Level	No			
Balance To Balance				
Account to Account				
Recharge	No	Subscriber doing a recharge	Notifications are generated when the successful recharge is done.	
SS_Change				
Award	No	Award set to true. Make a voice call.	Notifications are generated when the award is granted.	
NearAward	No	NearAward set to true. Make a voice call.	Notifications are generated when the Near Award is granted.	
Recurring Charge Full	No			
Recurring Charge No Charge	No			
Recurring Charge Partial Full	No			
PO Change	No		Notifications are generated when the successful PO swap is done.	
NRC No Charge	No			
NRC Charge Full	No			
BalDeferred Grant Activated(N-1)	No	Deferred Grant activated.	Notifications are generated when the Deferred Grant is activated.	
BalGrant Expiry NearExpire(1-N)	Yes	Grant expiry date is compared with the system date.	See * below table.	
BalGrantExpired(1-N)	No	Grant is expired	Notifications are generated when the Grant is expired.	

Example: The Grant expiry date is 06 June 2008 for core balance. System current date is 01 June 2008. Grant trigger configured as BalGrantNearExpire [core] <6. With this example, the following table applies:

Table 15 BalGrantExpiryNearExpire Example Results

System Date	Latch Off	Latch On
1-Jun-08	Notification generated	Notification generated
2-Jun-08	Notification generated	Notification not generated
3-Jun-08	Notification generated	Notification not generated
4-Jun-08	Notification generated	Notification not generated
5-Jun-08	Notification generated	Notification not generated
6-Jun-08	Notification generated	Notification not generated

Latching Compound Triggers

For compound triggers generation, all conditions must be true. However, latched compound triggers fire again if one of the conditions has been false because of the initial notification (OR CONDITION?).

Tokens

A token is a value that is assigned to a primary offer and triggers a notification to the subscriber regarding the value. The option exists to insert more than one token into a message. There is no limit to the number of tokens inserted into a message. This is limited by the interface message size.

Smart Tokens

Smart Tokens is a mechanism providing the capability to suppress messages if the value of a balance is zero. With Smart Tokens, network operators choose to suppress notifications if a balance contains a zero value or null values. The following rules apply to Smart Tokens:

- Messages contain Smart Tokens delimited by "{" and "}". A sample message is "Hi Dan. {Your balance is ~Balance[2]~./Zero/} Thanks!". The entire smart token is "{Your balance is ~Balance[2]~./Zero/}". The /Zero/ delimiter (or rule indicator) prepares us for the future phase requirements to support other rules (/Negative/) and indicates that this is the zero/null suppression rule.
- The entire Smart Token is suppressed if any of the tokens contained within the delimiters { and } is zero or null. For example, if the entire message is "Hi Dan. {Your balance is ~Balance[2]~ and ~Balance[3]~./Zero/} Your balance 4 is ~Balance[4]~. Thanks!" and either Balance[2] or Balance[3] is zero, then the entire Smart Token is suppressed and the subscriber sees the message: "Hi Dan. Your balance 4 is 432.34. Thanks!"
- If the tokens have values, they are replaced and the delimiters { and } are removed along with the rule /Zero/. Using the above example, if Balance[2] and Balance[3] do have values, then the subscriber sees the message: "Hi Dan. Your balance is 100.45 and 321.22. Your balance 4 is 432.34. Thanks!"
- If suppression of a Smart Token results in an empty notification then that notification is not sent.
- Multiple Smart Tokens that occur in a single message are not nested.
- The PC is not altered. Smart Tokens are typed in manually by the operator.

Token Selection and Definition

The PC provides a list of valid tokens. If a token (value) is to be associated with the notification message, a selection is made from the provided dropdown list.

Selecting a token from the list, within the context of the message text, directs the system to insert the value for that token into the message.

When defining notification messages:

- The message contains any text.
- The message includes subscriber-specific or account-specific information, such as the subscriber's current balance or expiration date. Such fields are stored as predefined tokens and are replaced by the actual subscriber data when the message is sent.
- The tokens, such as the subscriber ID, are used in the messages to let the HLR know which subscriber is affected by the command (for example, barred or enabled).
- For example, a message text is "You have $~balance~ in your account", where the ~balance~ token represents the subscriber balance. When used for a subscriber with a balance of $20.00, the actual SMS message is "You have $20.00 in your account."

The following table lists all the available token types, token names, details regarding the announcement and whether the token applies to subscriber, account or both.

Table 16 Tokens Available

Token Type	Token	Announcement Value	Subscriber, Account or Both
Subscriber Notification Token (subscriber) Accumulator Token (account)	AccumulatorValue(1-N)	Announces accumulator values for Accumulators 1-N. Value is rounded to T decimal places where T is configurable on a system-wide basis up to six digits. This token is displayed on the screen as ~ AccumulatorValue[Accu1]~ ~ AccumulatorValue[Accu2]~	Both
Subscriber Notification Token (subscriber) Accumulator Token (account)	AccumulatorName(1-N)	Announces the accumulator name. This token is displayed on the screen as ~ AccumulatorName[Accu1]~ ~ AccumulatorName[Accu2]~	Both
Subscriber Notification Token (subscriber) Accumulator Token (account)	PlanAccumThreshold	This is specific to the threshold value that caused the notification of the award. This token is only used with the specific promotion (near award/award events) and notifies on primary accumulator. This token is displayed on the screen as ~PlanAccumThreshold~	Both
Accumulator Token	PlanAccumThreshold1	This is specific to the threshold value that caused the notification of the award. This token is only used with the specific promotion (near award/award events) and notifies on the second accumulator (the first add-on accumulator). This token is displayed on the screen as ~PlanAccumThreshold1~	Both
Accumulator Token	PlanAccumThreshold2	This is specific to the threshold value that caused the notification of the award. This token is only used with the specific promotion (near award/award events) and notifies on the second accumulator (the third add-on accumulator). This token is displayed on the screen as ~PlanAccumThreshold2~	Both

Table 16 Tokens Available (Continued)

Token Type	Token	Announcement Value	Subscriber, Account or Both
Accumulator Token	PlanAccumThreshold3	This is specific to the threshold value that caused the notification of the award. This token is only used with the specific promotion (near award/award events) and notifies on the second accumulator (the fourth add-on accumulator). This token is displayed on the screen as ~PlanAccumThreshold3~	Both
Accumulator Token	PlanAccumThreshold4	This is specific to the threshold value that caused the notification of the award. This token is only used with the specific promotion (near award/award events) and notifies on the second accumulator (the fifth add-on accumulator). This token is displayed on the screen as ~PlanAccumThreshold4~	Both
Subscriber Notification Token (subscriber) Accumulator Token (account)	BalLimitType(1-N)	Announces the balance limit type for the account/subscriber spending or credit limit. This token is displayed on the screen as ~BalLimitType(1-N)~	Both
Subscriber Notification Token (subscriber) Accumulator Token (account)	BalLimitResetValue(1-N)	Announces the balance limit reset value. This token is displayed on the screen as ~BalLimitResetValue[Bal1]~ ~BalLimitResetValue[Bal2]~	Both
Subscriber Notification Token (subscriber) Accumulator Token (account)	BalLimitResetDate(1-N)	Announces the date for the next credit reset. This token is displayed on the screen as ~ BalLimitReset Date[Bal1]~ ~ BalLimitResetDate[Bal2]~	Both
Subscriber Notification Token (subscriber) Accumulator Token (account)	BalanceType(1-N)	Announces the balance type (prepaid, postpaid, shadow). This token is displayed on the screen as ~ BalanceType [Bal1]~, ~ BalanceType [Bal2]~	Both
Subscriber Notification Token (subscriber) Accumulator Token (account)	BalanceMinimum(1-N)	Announces the balance minimum. If the same balance exists in different offers, announces the minimum of the minimum. Uses balance minimum provisioned at offer(s) ref level. This token is displayed on the screen as ~ BalanceMinimum [Bal1]~, ~ BalanceMinimum [Bal2]~	Both
Subscriber Notification Token (subscriber) Accumulator Token (account)	BalanceMaximum(1-N)	Announces the balance maximum. In the case of same balance in different offer, announces max of max. Uses balance max provisioned at offer(s) ref level. ~ BalanceMaximum [Bal1]~, ~ BalanceMaximum [Bal2]~	Both
Subscriber Notification Token (subscriber) Accumulator Token (account)	TMLMax	Announces the TML max value. This token is displayed on the screen as ~ TMLMax~	Both

Table 16 Tokens Available (Continued)

Token Type	Token	Announcement Value	Subscriber, Account or Both
Subscriber Notification Token (subscriber) Accumulator Token (account)	TMLDelta	Announces the TML change/delta due to any activity and is used with an activity charge event. (Equals the difference between TMLLimit and the TMLValue in the account_subscriber table.)	Both
Subscriber Notification Token (subscriber) Accumulator Token (account)	TMLValue	Announces the TML current value. This token is displayed on the screen as ~ TMLValue~	Both
Subscriber Notification Token	ShadowBalLimit(1-N)	Announces the lower amount of either the Shadow Balance Value or the available funds in the Target Balance. This token is displayed on the screen as ~ ShadowBalLimit~	Subscriber
Subscriber Notification Token (subscriber) Accumulator Token (account)	MSISDN	Token for each External Id associated with subscriber. This token is displayed on the screen as ~MSISDN~	Both
Subscriber Notification Token	SSName	Announces the current shadow subscriber name. This token is displayed on the screen as ~SSName~	Subscriber
Subscriber Notification Token (subscriber) Accumulator Token (account)	SumCurrBalances(1-N)	Announces the Sum of All Currency Balances. This token is displayed on the screen as ~ SumCurrBalances [Currency1]~, ~ SumCurrBalances[currency2]~	Both
Subscriber Notification Token (subscriber) Accumulator Token (account)	Offer (1-N)	Announces the supplementary offer name in the chosen language substitution. This token is displayed on the screen as ~ OFFER [offer1]~, ~ OFFER[offer2]~	Both
Subscriber Notification Token (subscriber) Accumulator Token (account)	Balance (1-N)	Announces balance value as follows: Currency Balances: rounded to T decimal places where T is configurable on a system-wide basis up to six digits. Non-currency Balances and Charges (USSD and SMS): rounded to one decimal place. This token is displayed on the screen as ~ OFFER [offer1]~, ~ OFFER[offer2]~	Both

Table 16 Tokens Available (Continued)

Token Type	Token	Announcement Value	Subscriber, Account or Both
Subscriber Notification Token (subscriber) Accumulator Token (account)	Balance (Smart_ Seconds)	Announces “X minutes and Y seconds” where X and Y are the equivalent minutes and seconds of the Seconds balance. Supports duration (seconds) balances only. Format is <Smart_Seconds> If the value=0 then the token expands to: 0. If the value is <3600 but greater than 0 then the token expands to: (MM:SS) – XX:YY where XX and YY are the values in minutes and seconds. If the value is > or = 3600 then the token expands to: (HH:MM) – XX:YY where XX and YY are the values in hours and minutes (seconds are truncated). XX could be XXX, XXXX, and so on. Refer to the Free Second FDD for details.	Both
Subscriber Notification Token (subscriber) Accumulator Token (account)	BalanceMajor	Announces balance major (1-N) in whole numbers. For example, if Core balance is 10.55, announces “10”. This token is displayed on the screen as ~ BalanceMajor [Bal]~, ~ BalanceMajor [Bal2]~	Both
Subscriber Notification Token (subscriber) Accumulator Token (account)	BalanceMinor	Announces fractional balances. For example, if the Core balance is 10.55, announces “.55”. This token is displayed on the screen as ~ BalanceMinor [Bal]~, ~ BalanceMinor [Bal2]~	Both
Subscriber Notification Token (subscriber) Accumulator Token (account)	PlanName (1-N)	Announces the promotion Plan Name. Applies to discount and bonus items. This token is displayed on the screen as ~ PlanName [Promo1]~, ~ PlanName [Promo2]~	Both
Subscriber Notification Token (subscriber) Accumulator Token (account)	Charge (1-N)	Currently referred to as Charge and announces the amount charged to the Core balance for an activity (at the end of the activity when it’s actually charged). Charge is the charge amount for a given balance. The Charge token requires a balance specification (~Charge[CORE]~).	Both
Subscriber Notification Token	POName	Announces subscriber’s current Primary Offer Name. This token is displayed on the screen as ~ POName~	Subscriber
Subscriber Notification Token (subscriber) Accumulator Token (account)	CurrencyCharge	CurrencyCharge announces the sum of the charge amounts for all of the currency balances in use for the current activity. This token is displayed on the screen as ~ CurrencyCharge~	Both
Subscriber Notification Token (subscriber) Accumulator Token (account)	ActivityOriginatorID	Announces the activity originator MSISDN. This token is displayed on the screen as ~ ActivityOriginator~	Both

Table 16 Tokens Available (Continued)

Token Type	Token	Announcement Value	Subscriber, Account or Both
Subscriber Notification Token (subscriber) Accumulator Token (account)	BNumber	Announces the called party number (B number) MSISDN. Only used with activity characteristic events. This token is displayed on the screen as ~ BNumber~	Both
Subscriber Notification Token (subscriber) Accumulator Token (account)	ExpireDate	Announces the balance expiration date. This token is displayed on the screen as ~ ExpireDate[Bal1]~ , ~ ExpireDate[Bal2]~	Both
Subscriber Notification Token (subscriber) Accumulator Token (account)	IMSI	Announces subscriber IMSI. This token is displayed on the screen as ~ IMSI~	Both
Subscriber Notification Token (subscriber) Accumulator Token (account)	RewardNotifText	Announces the reward notification text as provisioned on the bonus/discount windows. Reward notification text allows up to 160 characters.	Both
Subscriber Notification Token (subscriber) Accumulator Token (account)	State	Announces subscriber state.	Both
Subscriber Notification Token (subscriber) Accumulator Token (account)	SubscriberId	Announces subscriber ID (MSISDN) for backward compatibility.	Both
Subscriber Notification Token	AffectedSubscriber	Announces newly added, disconnected or transferred out subscriber ID or SSName. DR-4-006-138	Account
General Token	Smart Token	Used to suppress zero and null value messages.	Both
General Token	PREVSTATE	Subscriber's previous state	Subscriber
General Token	EventDate_MM	Announces the month in the subscriber's timezone.	Both
General Token	EventDate_DD	Announces the date in the subscriber's timezone.	Both
General Token	EventDate_YYYY	Announces the year in the subscriber's timezone.	Both
General Token	EventTime_HH	Announces the hour in the subscriber's timezone.	Both
General Token	EventTime_MIN	Announces the minutes in the subscriber's timezone.	Both
General Token	EventTime_SEC	Announces the seconds in the subscriber's timezone.	Both
General Token	TMLDelta	Announces the TML change/delta due to any activity and is used with an activity charge event. (Represents the difference between the TMLLimit and the TMLValue in the account_subscriber table.)	Account
General Token	SSID	Shadow subscriber ID	Subscriber
General Token	SSNAME	Shadow subscriber name.	Subscriber

Table 16 Tokens Available (Continued)

Token Type	Token	Announcement Value	Subscriber, Account or Both
Promised Payment tokens (used only by DB)	PPAMT	Promised payment amount.	Subscriber
Promised Payment tokens (used only by DB)	PPSERVFEE	Promised payment service fee	Subscriber
Promised Payment tokens (used only by DB)	PPDUEDT	Promised payment due date	Subscriber
Promised Payment tokens (used only by DB)	SUBBAL	Subscriber balance	Subscriber
Promised Payment tokens (used only by DB)	TOTPPAMT	Total promised payment amount	Subscriber
Daily Fee with no activity (used only by DB)	Daily_Fee_Deducted	Daily fee deducted	Subscriber
Accumulator Token	AccountNo		Both
Accumulator Token	EventDate_MM	Announces the month in subscriber timezone.	Both
Accumulator Token	EventDate_DD	Announces the date in subscriber timezone.	Both
Accumulator Token	EventDate_YYYY	Announces the year in subscriber timezone.	Both
Accumulator Token	EventTime_HH	Announces the hour in subscriber timezone.	Both
Accumulator Token	EventTime_MIN	Announces the minutes in subscriber timezone.	Both
Accumulator Token	EventTime_SEC	Announces the seconds in subscriber timezone.	Both
Balance Transfer Notification Token (subscriber) Accumulator Token (account)	BalTransAmt	Announces transfer amount. Follows the same precision as balances above. This token is displayed on the screen as ~ BalTransAmt~	Both
Balance Transfer Notification Token (subscriber) Accumulator Token (account)	DestBalName	Announces destination balance name. This token is displayed on the screen as ~ DestBalName~	Both
Balance Transfer Notification Token (subscriber) Accumulator Token (account)	RecipientSubId	Announces the recipient subscriber ID. This token is displayed on the screen as ~ ReceipientSubID~	Both

Table 16 Tokens Available (Continued)

Token Type	Token	Announcement Value	Subscriber, Account or Both
Balance Transfer Notification Token (subscriber) Accumulator Token (account)	BalTransOriginator	Announces the balance transfer originator ID (MSISDN). This token is displayed on the screen as ~ BalTransOriginator~	Both
Converged Notification Token	OwningAccountID	Announces the owning account ID (account_no). This token is displayed on the screen as ~ OwningAccountID~	Both
Converged Notification Token	PrevState	Announces the subscriber's immediate previous state, if any. This token is displayed on the screen as ~ PrevState~	Subscriber
Voucher Notification Token	ChangeInAccum	Announces the delta value due to recharge. This token is displayed on the screen as ~ ChangeInAccum[Accum1]~ ~ ChangeInAccum[Accum2]~	Both
Voucher Notification Token	ChangeInBal	Announces the balance delta value. For example, the card face value is $40 but the actual amount deposited in the subscriber balance is $50. This token announces "50". This token is displayed on the screen as ~ ChangeInBal[Bal1]~, ~ ChangeInBal[Bal2]~	Both
Voucher Notification Token	VoucherBatchNumber	Announces the voucher batch number in which this card was created. This token is displayed on the screen as ~VoucherBatchNumber~	Both
Voucher Notification Token	VoucherFaceValue	Announces the voucher face value used for recharge. This token is displayed on the screen as ~VoucherFaceValue~	Both
Voucher Notification Token	VoucherOffsetValue	Announces the balance offset value (days) from the recharge control table. This token is displayed on the screen as ~VoucherOffsetValue~	Both
Voucher Notification Token	VoucherSerialNumber	Announces the voucher serial number. This token is displayed on the screen as ~VoucherSerialNumber~	Both
Recurring Charge / Nonrecurring Charge SMS Token	RCAwardBalAmount	Announces charge award amounts for non-Core balances. This can be multiple balances since there is a token for each balance award. This token is displayed on the screen as ~ RCAwardBalAmount[bal1]~	Both
Recurring Charge / Nonrecurring Charge SMS Token	RCNRCChargeAmount	Announces the total amount charged to all of the balances. This token is displayed on the screen as ~RCNRCChargeAmount~	Both
Recurring Charge / Nonrecurring Charge SMS Token	RCNRCChargeName	Announces the name of the charge. This token is displayed on the screen as ~RCNRCChargeName~	Both

Table 16 Tokens Available (Continued)

Token Type	Token	Announcement Value	Subscriber, Account or Both
Recurring Charge / Nonrecurring Charge SMS Token	RCNRCDelta	Announces the amount not covered by the available balance. This token is displayed on the screen as ~RCNRCDelta~	Both
Subscriber USSD / FeatReq Token	Bal(,|.)	Balance Format: ~Balance(,|.)~: where “,” is the thousand separator; “.” defines the decimal point. For example: 1,200.45 VoucherValue Format: ~Balance(,|.)~: where “,” is the thousand separator; “.” defines the decimal point. For example: 1,200.45	Subscriber
Subscriber USSD / FeatReq Token	Bal(|.)	Balance Format: ~Balance(,|.)~: where “,” is the thousand separator; “.” defines the decimal point. For example: 1,200.45 VoucherValue Format: ~Balance(,|.)~: where “,” is the thousand separator; “.” defines the decimal point. For example: 1,200.45	Subscriber
Subscriber USSD / FeatReq Token	Bal(,|)	Balance Format: ~Balance(,|.)~: where “,” is the thousand separator; “.” defines the decimal point. For example: 1,200.45 VoucherValue Format: ~Balance(,|.)~: where “,” is the thousand separator; “.” defines the decimal point. For example: 1,200.45	Subscriber
Subscriber USSD / FeatReq Token	Bal(.|,)	Balance Format: ~Balance(,|.)~: where “,” is the thousand separator; “.” defines the decimal point. For example: 1,200.45 VoucherValue Format: ~Balance(,|.)~: where “,” is the thousand separator; “.” defines the decimal point. For example: 1,200.45	Subscriber
Subscriber USSD / FeatReq Token	Bal(Smart_Secs)	Balance (Smart_Seconds) announces “X minutes and Y seconds” where X and Y are the equivalent minutes and seconds of the Seconds balance. Supports duration (seconds) balances only. Format is <Smart_Seconds> 1. If the value=0 then the token expands to: 0. 2. If the value is <3600 but greater than 0 then the token expands to: (MM:SS) – XX:YY where XX and YY are the values in minutes and seconds. 3. If the value is > or = 3600 then the token expands to: (HH:MM) – XX:YY where XX and YY are the values in hours and minutes. (seconds are truncated). Note that XX could be XXX, XXXX, etc.	Subscriber
Subscriber USSD / FeatReq Token	CurrencyType		Subscriber
Subscriber USSD / FeatReq Token	DD	~DD~ Subscriber's day of expiration in DD format	Subscriber
Subscriber USSD / FeatReq Token	ExpireDate		Subscriber
Subscriber USSD / FeatReq Token	YYYY	~YYYY~ Subscriber's year of expiration in YYYY format	Subscriber

Table 16 Tokens Available (Continued)

Token Type	Token	Announcement Value	Subscriber, Account or Both
Subscriber USSD / FeatReq Token	MM	~MM~ Subscriber's month of expiration in MM format	Subscriber
Subscriber USSD / FeatReq Token	YY	~YY~ Subscriber's year of expiration in YY format	Subscriber
Subscriber USSD / FeatReq Token	State		Subscriber
Subscriber USSD / FeatReq Token	AllCurBal		Subscriber
Subscriber USSD / FeatReq Token	BalDelta(|,)	Balance Format: ~Balance(,|.)~: where “,” is the thousand separator; “.” defines the decimal point. For example: 1,200.45 VoucherValue Format: ~Balance(,|.)~: where “,” is the thousand separator; “.” defines the decimal point. For example: 1,200.45	Subscriber
Subscriber USSD / FeatReq Token	BalDelta(|.)	Balance Format: ~Balance(,|.)~: where “,” is the thousand separator; “.” defines the decimal point. For example: 1,200.45 VoucherValue Format: ~Balance(,|.)~: where “,” is the thousand separator; “.” defines the decimal point. For example: 1,200.45	Subscriber
Subscriber USSD / FeatReq Token	BalDelta(,|.)	Balance Format: ~Balance(,|.)~: where “,” is the thousand separator; “.” defines the decimal point. For example: 1,200.45 VoucherValue Format: ~Balance(,|.)~: where “,” is the thousand separator; “.” defines the decimal point. For example: 1,200.45	Subscriber
Subscriber USSD / FeatReq Token	BalDelta(.|,)	Balance Format: ~Balance(,|.)~: where “,” is the thousand separator; “.” defines the decimal point. For example: 1,200.45 VoucherValue Format: ~Balance(,|.)~: where “,” is the thousand separator; “.” defines the decimal point. For example: 1,200.45	Subscriber
Subscriber USSD / FeatReq Token	BalType		Subscriber
Subscriber USSD / FeatReq Token	LimitType		Subscriber
Subscriber USSD / FeatReq Token	LimitVal(|,)		Subscriber
Subscriber USSD / FeatReq Token	LimitVal(|.)		Subscriber
Subscriber USSD / FeatReq Token	LimitVal(,|.)		Subscriber
Subscriber USSD / FeatReq Token	LimitVal(.|,)		Subscriber
Subscriber USSD / FeatReq Token	LrAvlId		Subscriber
Subscriber USSD / FeatReq Token	LrAvlName		Subscriber

Table 16 Tokens Available (Continued)

Token Type	Token	Announcement Value	Subscriber, Account or Both
Subscriber USSD / FeatReq Token	LrId		Subscriber
Subscriber USSD / FeatReq Token	LrName		Subscriber
Subscriber USSD / FeatReq Token	ResetDate		Subscriber
Voucher USSD / FeatReq Token	VoucherValue(,|.)		Subscriber
Voucher USSD / FeatReq Token	VoucherValue(|.)		Subscriber
Voucher USSD / FeatReq Token	VoucherValue(,|)		Subscriber
Voucher USSD / FeatReq Token	VoucherValue(.|,)		Subscriber

The following table provides information on the notifications tokens allowed per event and the exceptions for each event.

Table 17 Tokens Allowed per Event

Event	Allowed Token Types	Exceptions

State	Subscriber
	Converged
		Subscriber.Offer(1-N)
		Subscriber.PlanName(1-N)
		Subscriber.Charge(1-N)
		Subscriber.CurrencyCharge
		Subscriber.BNumber
		Subscriber.RewardNotifText

Balance	Subscriber
	Converged
		Subscriber.Offer(1-N)
		Subscriber.PlanName(1-N)
		Subscriber.Charge(1-N)
		Subscriber.CurrencyCharge
		Subscriber.BNumber
		Subscriber.RewardNotifText

Balance Expiration Date	Subscriber
	Converged
		Subscriber.Offer(1-N)
		Subscriber.PlanName(1-N)
		Subscriber.Charge(1-N)
		Subscriber.CurrencyCharge
		Subscriber.BNumber
		Subscriber.RewardNotifText

Previous State	Subscriber
	Converged
		Subscriber.Offer(1-N)
		Subscriber.PlanName(1-N)
		Subscriber.Charge(1-N)
		Subscriber.CurrencyCharge
		Subscriber.BNumber
		Subscriber.RewardNotifText

Table 17 Tokens Allowed per Event (Continued)

Event	Allowed Token Types	Exceptions

Activity Characteristic	Subscriber
	Converged
		Subscriber.Offer(1-N)
		Subscriber.PlanName(1-N)
		Subscriber.RewardNotifText

Activity Charge	Subscriber
	Converged
		Subscriber.Offer(1-N)
		Subscriber.PlanName(1-N)
		Subscriber.RewardNotifText

Sub Notif Level	N/A - This is only a qualifying trigger.	N/A - This is only a qualifying trigger.

Balance To Balance	Subscriber
	Converged
	Balance Transfer
		Subscriber.Offer(1-N)
		Subscriber.PlanName(1-N)
		Subscriber.Charge(1-N)
		Subscriber.CurrencyCharge
		Subscriber.BNumber
		Subscriber.RewardNotifText

Account To Account	Subscriber
	Converged
	BalanceTransfer
		Subscriber.Offer(1-N)
		Subscriber.PlanName(1-N)
		Subscriber.Charge(1-N)
		Subscriber.CurrencyCharge
		Subscriber.BNumber
		Subscriber.RewardNotifText

SS_Change	Subscriber
	Converged
		Subscriber.Offer(1-N)
		Subscriber.PlanName(1-N)
		Subscriber.Charge(1-N)
		Subscriber.CurrencyCharge
		Subscriber.BNumber
		Subscriber.RewardNotifText

Award	Subscriber
	Converged
	Grant
		Subscriber.Offer(1-N)
		Subscriber.PlanName(1-N)
		Subscriber.Charge(1-N)
		Subscriber.CurrencyCharge
		Subscriber.BNumber

Near Award	Subscriber
	Converged
	Grant
		Subscriber.Offer(1-N)
		Subscriber.PlanName(1-N)
		Subscriber.Charge(1-N)
		Subscriber.CurrencyCharge
		Subscriber.BNumber

Recharge	Subscriber
	Converged
	Voucher
		Subscriber.Offer(1-N)
		Subscriber.PlanName(1-N)
		Subscriber.Charge(1-N)
		Subscriber.CurrencyCharge
		Subscriber.BNumber
		Subscriber.RewardNotifText

Recurring Charge Full	Subscriber
	Converged
	RecurringCharge
	Grant
		Subscriber.Offer(1-N)
		Subscriber.PlanName(1-N)
		Subscriber.Charge(1-N)
		Subscriber.CurrencyCharge
		Subscriber.BNumber
		Subscriber.RewardNotifText

Table 17 Tokens Allowed per Event (Continued)

Event	Allowed Token Types	Exceptions

Recurring Charge No Charge	Subscriber
	Converged
	RecurringCharge
	Grant
		Subscriber.Offer(1-N)
		Subscriber.PlanName(1-N)
		Subscriber.Charge(1-N)
		Subscriber.CurrencyCharge
		Subscriber.BNumber
		Subscriber.RewardNotifText

Recurring Charge Partial Full	Subscriber
	Converged
	RecurringCharge
	Grant
		Subscriber.Offer(1-N)
		Subscriber.PlanName(1-N)
		Subscriber.Charge(1-N)
		Subscriber.CurrencyCharge
		Subscriber.BNumber
		Subscriber.RewardNotifText

PO Change	Subscriber Converged	Subscriber.PlanName(1-N) Subscriber.Charge(1-N) Subscriber.CurrencyCharge Subscriber.BNumber Subscriber.RewardNotifText

NRC No Charge	Subscriber
	Converged
	RecurringCharge
		Subscriber.Offer(1-N)
		Subscriber.PlanName(1-N)
		Subscriber.Charge(1-N)
		Subscriber.CurrencyCharge
		Subscriber.BNumber
		Subscriber.RewardNotifText

NRC Charge Full	Subscriber
	Converged
	RecurringCharge
		Subscriber.Offer(1-N)
		Subscriber.PlanName(1-N)
		Subscriber.Charge(1-N)
		Subscriber.CurrencyCharge
		Subscriber.BNumber
		Subscriber.RewardNotifText

BalDeferred Grant Activated(1-N)	Subscriber
	Converged
	Grant
		Subscriber.Offer(1-N)
		Subscriber.PlanName(1-N)
		Subscriber.Charge(1-N)
		Subscriber.CurrencyCharge
		Subscriber.BNumber
		Subscriber.RewardNotifText

BalGrantExpired(1-N)	Subscriber
	Converged
	Grant
		Subscriber.Offer(1-N)
		Subscriber.PlanName(1-N)
		Subscriber.Charge(1-N)
		Subscriber.CurrencyCharge
		Subscriber.BNumber
		Subscriber.RewardNotifText

Data Byte Balances

Large data balances may be represented in kilobytes, megabytes, gigbytes, and terabytes for notifications and USSD responses. Smart Tokens are supported which provide automatic conversion of bytes balance to kilobytes, megabytes, gigabytes or terabytes based on a range to which the value of the balance belongs. The number of digits to the right of the decimal may vary for each type of byte and is defined by service parameters. These values are accurate and precise when applied to noncurrency balance bytes. These tokens are not applicable for currency balances and duration/seconds balances. The only validation supported when applying these tokens is to disallow application of these tokens to currency balances and seconds balances.

NOTE The operator configures the bytes tokens in the PC at the Account and Primary Offer level.

The following table defines the Unit Values Representing Bytes tokens for notifications and USSD balances.

Table 18 Unit Values Representing Bytes Tokens

Byte	Notification	USSD Responses
kilobyte	Balance(KBytes)	Bal(KBytes)
megabyte	Balance(MBytes)	Bal(MBytes)
gigabyte	Balance(GBytes)	Bal(GBytes)
terabyte	Balance(TBytes)	Bal(TBytes)
kilobyte	Balance(SmartKBytes)	Bal(SmartKBytes)
megabyte	Balance(SmartMBytes)	Bal(SmartMBytes)
gigabyte	Balance(SmartGBytes)	Bal(SmartGBytes)
terabyte	Balance(SmartTBytes)	Bal(SmartTBytes)

NOTE When provisioning notifications and USSD responses, the tokens have meaning only when applied to unit values that represent bytes. It is possible to misprovision these tokens to other unit values, such as SMS or MMS. This may result in presenting incorrect values in USSD responses and notifications.

Notification Language

In the PC, the Comverse ONE solution provides a dropdown list of all system-supported languages. Once a message is defined, any of the provided languages is selected.

NOTE If the language on the subscriber details panel is different from the primary offer notification language, no notification is sent. Notification languages must match otherwise the system suppresses the notification. For each trigger on the PO, multiple language messages must be defined if different subscribers on the PO desire their notifications in different languages. The system does not translate notification languages and PO default language is not used in notification. Translations are the responsibility of the customer.

Many languages are supported simultaneously including Hebrew, Cyrillic, and Spanish. Through the CC, a subscriber requests a different notification language from the primary offer default language.

NOTE

Specific to non-voucher recharges, if the description is not available in the Base Language for the Subscriber's Notification Language, the description from the Default Language (as defined in the System Parameter CDR_NOTIF_LANGUAGE) is used.

Notification messages are entered in the PC. The handling of tokens is not affected by mixed languages. If the notification message contains languages other than English, the system recognizes this and the Unicode flag is set as true in the po_NOTIF or po_USSD_RESPONSE tables.

No new roles are required. Any PC user with the AR_SERVICE PROVISIONING role has read and modify privileges to the primary offer USSD response and primary offer USSD Callback tabs.

NOTE

Changing the USSD language does not change the IVR language. Each language must be changed separately.

Additional Language Support

The USSD Responses (InfoServer, USSD Recharge Server, USSD Callback, and error messages) are also required to support Cyrillic characters using the Unicode (as UTF-8) standard characters. Again, given the fact that the UTF-8 Unicode representation is equivalent to the ASCII representation for English characters, the message parsing remains unaffected.

Whenever notifications are sent, the device receiving the notifications is informed that the message is in the UTF-8 Unicode format when applicable. In the case of SMS, the SMS Gateway is configured to use UTF-8 encoding. It then assumes that every message it receives is in the UTF-8 format.

For USSD, the message sent to IPF (IpsnUssdMessage) also contains an Alphabet Id. The Alphabet Id distinguishes the character encoding used (default, 8-bit, or 8-bit for UTF-8).

A notification message in any non-Asian language [Rciril-005] entered into Notification / USSD / USSD Callback textbox is parsed in UTF-16 format for UTF-8 conversion. UTF-8 encoded [Rcyril-002.1] [Rcyril-003] text is packed as pairs of octets into byte characters accordingly, resulting in a readable, matching ANSI format for English characters and unreadable symbols for any other language. Maximum number of octets supported is 160.

Unified Application Programming Interface (UAPI)

For any balance change request from UAPI to set a balance value or to add funds that results in higher value than the maximum value allowed for the balance, the Comverse ONE solution rejects the request and warns the operator with an appropriate message.

For CC/UAPI activities that results in event (balance, spending limit, and so on) changes, the Comverse ONE solution generates a notification to the subscriber if the event condition crosses a defined threshold.

From a notification point of view, the service provider representative has three tasks to perform via CC or UAPI interface:

- Assign subscriber to a primary offer that has notifications
- Set Subscriber Notification Language
- Set Sub_Notif_level for each subscriber

Refer to the Product Catalog User Guide for details regarding provisioning UAPI notifications.

Index