



Application Help

SAP Convergent Charging (SAP CC 5.0)

Target Audience

- Solution and Technology Consultants
- Implementation Project Teams
- Application and System Administrators
- Operation Team Experts
- Support Specialists

PUBLIC

Document version: 1.4 – May 2019

Document History



CAUTION

Before you start the implementation, make sure you have the latest version of this document. You can find the latest version at the following locations: <http://help.sap.com/cc50>.

The following table provides an overview of the most important document changes.

Version	Date	Description
1.0	May 2017	Initial version (5.0 SP 0)
1.1	November 2017	First maintenance version (5.0 SP 1)
1.2	May 2018	Second maintenance version (5.0 SP 2)
1.3	November 2018	Third maintenance version (5.0 SP 3)
1.4	May 2019	Third maintenance version (5.0 SP 4)

Table of Contents

DOCUMENT HISTORY	2
TABLE OF CONTENTS	3
1 BEFORE YOU START.....	11
Target Audience.....	11
Fundamental User Assistance and Guides.....	11
Additional Information	11
2 GETTING STARTED	12
About this Document.....	12
What's New in This Document	12
<i>What's New in SP 4</i>	<i>12</i>
<i>What's New in SP 3</i>	<i>13</i>
<i>What's New in SP 2</i>	<i>13</i>
<i>What's New in SP 1</i>	<i>13</i>
Document Abbreviations.....	13
3 OVERVIEW.....	16
Presentation	16
Technical System Landscape	18
4 FEATURES.....	20
Product Modeling	20
Order Management.....	20
<i>Enterprise Credit Pooling.....</i>	<i>20</i>
Allowance Management.....	20
<i>Allowance Modeling.....</i>	<i>20</i>
<i>Allowance Usage Modeling.....</i>	<i>21</i>
<i>Limitation</i>	<i>21</i>
Chargeable Items Acquisition	21
Online Charging	22
Offline Charging	24
Deferred Charging.....	25
Manual Capturing	26
Data Files Bulk Loading	27
Chargeable Items Rerating	27
<i>Rerating Operations driven by the BART Server.....</i>	<i>28</i>
<i>Rerating Operations driven by SAP Convergent Invoicing.....</i>	<i>29</i>
Prepaid Accounts Refilling	30
Tax Calculation.....	30
<i>Preconfigured EU VAT Rules.....</i>	<i>31</i>
Charged Items Aggregation	32
Real-Time Credit-Control.....	37
Spending Status Monitoring	38

<i>Feature Configuration</i>	39
<i>Integration with a Policy or Monitoring System</i>	40
<i>Spending Status Reporting</i>	40
<i>Functions and Mechanism Details</i>	40
<i>Limitations</i>	41
Catalog Transport	41
<i>Transport Destinations</i>	41
<i>Transport Requests</i>	41
Transport requests handled by the SAP CC system	41
Transport requests handled by the SAP CTS system	44
Provider Contract Transfer	44
<i>Overview</i>	44
<i>Customer Data</i>	45
<i>Operational Status of a Provider Contract</i>	45
<i>Export and Import Operations</i>	46
High Availability	46
<i>Hardware High Availability</i>	46
<i>Software High Availability</i>	47
Heartbeat	47
Databases active-active high availability	47
Databases active-passive high availability	49
High Volume Management	50
<i>Routing Mechanisms</i>	50
Partitioning Routing	50
Round-robin routing	53
Multicast routing	53
<i>Caching Policies</i>	54
Configurable caches	54
Auto-sized caches	56
Cache warm-up	57
Network Data Transfer	57
Synthesis	58
<i>Batch rating group management</i>	59
System and Data Auditing	60
<i>Master Data and User Operation Auditing</i>	61
Recording of Object Change Logs	62
Recording of Object Snapshots	64
Recording of User Operations	66
Functions and Mechanism Details	67
Feature Implementation	68
<i>Product Usage Measurement</i>	68
Product Usage Metrics	69
GLAS Files	70
Customer Usage Measurement	70
Data Purging	70
<i>Allowances Purge</i>	71
<i>Change Lists Purge</i>	72
Enhanced Logging and Tracing Framework	72
<i>Logging</i>	73
Log Severities, Thresholds, and Filtering	73
Categories	75
<i>Tracing</i>	75
Trace Severities, Threshold, and Filtering	76
Tracing Domains	78
Performance Traces	78
Targeted Tracing	79

- Output Formats* 79
 - List Format 80
 - Trace Format 80
 - XML Format 81
- Output Destinations* 81
- Configuration* 81
- SAP Passport Handling* 82
 - Supported SAP Passports and Available Trace Flags 82
 - SAP Passport and End-to-End Tracing 85
 - SAP Passport and Business Process Tracing 86
- Logging and Tracing Framework of Diameter Server 87
 - Output Format* 87
 - Configuration* 87
- User Session Management 87
 - In Desktop Applications* 88
 - User Session Lifecycle 88
 - In Web Application* 88
- Table rollover for rating sessions 89
- License Management 89
 - Temporary SAP License* 90
 - Permanent SAP License* 90
 - SAP License Auditing* 90

5 ARCHITECTURE 91

- Synthetic Overview 91
- Components and Tools 93
 - Core Server* 93
 - Core Tool 94
 - Setup Tool 95
 - Config Tool 96
 - Cockpit 96
 - Admin+ 97
 - HTTP Client 97
 - Dispatcher 98
 - Guider 100
 - Rater 100
 - Updater 101
 - Taxer 102
 - Bulkloader 102
 - Synthesis 103
 - BART Server* 105
 - BART Tool 106
 - BART Setup Tool 106
 - BART+ 106
 - Diameter Server* 107
 - IEC* 111
 - Databases* 111
 - Core Database 111
 - Session Database 111
 - Cockpit Database 111
 - BART Database 112
 - IEC Database 112
 - Database technologies 112
 - Database editions 112
 - Database partitioning 112

6 MASTER DATA 118

Master Data for the Service Provider	119
<i>Product</i>	119
<i>Service Provider</i>	120
<i>Catalog</i>	120
<i>Chargeable Item</i>	120
Chargeable item class	121
Chargeable item package	121
<i>Charged Item</i>	121
Charged item class	121
Charged item mapping	121
<i>Billable item</i>	121
Billable item class	121
Billable item mapping	122
<i>Consumption item</i>	122
Consumption item class	122
Consumption item mapping	122
<i>Charge</i>	122
Master and dependent charges	123
Combination of charges	123
<i>Counter</i>	123
Persistent and transient counters	124
Counter redefinition	124
Counter sharing	124
Counter creation in a provider contract	125
Counter creation in a subscription	125
<i>Price Plan</i>	126
Price plan components	126
Counters	126
Parameters	127
Pricing macros	127
Translation tables	127
Tier tables	127
Mapping tables and mapping table classes	128
Price table	129
Range tables and range table classes	129
<i>Charging Plan</i>	130
Charging plan components	130
<i>Offer</i>	130
<i>Aggregation Policy</i>	131
<i>Charge Plan</i>	131
Combination of charge plans	132
Charges and dependencies	132
Counters and parameters	132
User Technical Identifiers	133
Expected Account Assignments	133
Versions	133
Lifecycle	133
Counter name dictionary	133
<i>Customized Charge</i>	133
Counters	134
Parameters	134
Tax Settings	134
Configuration of the generation of charged items	136
Configuration of the generation of response items	136
Charging reference mapping	136
Service ID declaration	136
<i>Refill Item</i>	136
Refill item class	136
<i>Refill Record</i>	137

Refill record class	137
<i>Refill Logic</i>	137
Components.....	137
<i>Refill Plan</i>	137
Combination.....	138
Counters and parameters	138
Counter sharing at provider contract level.....	138
Parameter redefinition at provider contract level.....	138
User Technical Identifiers.....	138
Expected account assignment.....	139
Versions	139
Lifecycle	139
Counter name dictionary	139
<i>Customized Refill Logic</i>	139
Counters.....	139
Parameters.....	139
Configuration of the generation of the refill records	140
Charging reference attributions	140
Service ID declaration.....	140
<i>Monitoring Plan</i>	140
Spending threshold settings.....	140
Versions	141
Lifecycle	141
Counter name dictionary	141
Spending status descriptions.....	141
<i>Allowance Event</i>	141
Allowance Event Class.....	142
<i>Allowance Logic</i>	142
Components.....	142
Persistent counters	143
Transient counters	143
Parameters.....	143
<i>Customized Allowance Logic</i>	143
<i>Allowance Plan</i>	143
Life cycle.....	144
Versions	144
<i>Allowance Interface</i>	144
Master Data for the Customer	145
<i>Access</i>	145
<i>Subscriber Account</i>	146
Tax settings	146
Prepaid balance.....	148
External account	148
Credit-limit balance.....	148
Alerts and notifications	149
<i>Business Data Tables</i>	149
Subscriber mapping tables.....	149
Subscriber range tables.....	149
<i>Subscription</i>	149
<i>Provider Contract</i>	150
Contract item	150
Activated plan (charge plans and a refill plan)	150
Shared counter.....	150
Version	151
<i>Contract Item</i>	152
Charge activation	152
Counter and counter share	152
Account assignment.....	152
Technical data and user technical identifiers	152

<i>Allowance</i>	153
Charge activation	153
Account Assignment.....	153
Currency.....	153
Allowance Validity Period.....	153
Counters.....	153
Parameters.....	154
Sharing Allowances Between Provider Contracts within a Subscriber Account	154
Sharing Allowances between Provider Contracts from separate Subscriber Accounts	154
BART Server Master Data	154
<i>CDR (Consumption Detail Record)</i>	154
CDR fields	155
CDR properties	156
Diameter Server Master Data.....	156
<i>CCR (Credit Control Request)</i>	157
<i>CCA (Credit Control Answer)</i>	158
7 PROCESSES AND FUNCTIONS	160
Activation Process.....	160
<i>Process Description</i>	160
Activation and Allowances	162
Activation and Linked Contracts.....	162
<i>External Triggering of the Activation Process</i>	162
<i>Internal Triggering of the Activation Process</i>	162
<i>Bulk Execution of the Activation Process</i>	163
<i>Activation Process and Dependent Charges</i>	163
<i>Transactions Generated by the Activation Process</i>	164
Product Modeling Process	164
<i>Process Description</i>	164
<i>Data Distribution</i>	165
<i>Cached Structures Refreshing</i>	165
<i>Process Limitations</i>	166
<i>Stateless Mode</i>	166
Data Files Bulk Loading Process.....	166
<i>Process Overview</i>	166
<i>Lock/Unlock Policy</i>	167
<i>Bulk Loading Mechanism</i>	167
Version 1 error files.....	169
Version 2 error files.....	169
<i>Configuration Parameters</i>	171
Order Management Process.....	171
<i>Process Description</i>	171
<i>Provisioning of Subscriber Account</i>	172
<i>Provisioning of Provider Contracts</i>	172
<i>Provisioning of Subscriptions</i>	173
<i>Provisioning of Accesses</i>	174
Prepaid Accounts Refilling Process.....	174
<i>Process Overview</i>	174
<i>External Events</i>	175
<i>Internal Events</i>	176
<i>Process Execution</i>	176
Chargeable Items Re-rating Process	178
<i>Process Overview</i>	178
<i>Process Limitations</i>	178

<i>Execution Modes</i>	179
Rerating driven by the BART Server	179
Rerating driven by the SAP ERP/FI-CA Component of the SAP Business Suite	183
Chargeable Items Charging Process	184
<i>Process Overview</i>	184
Precision & Rounding Modes	186
Charging Policy.....	186
Configuration Parameters	188
<i>Process Functions</i>	188
Rating Contexts Generation	188
Events Pricing	189
Events Charging.....	189
Allowance Creation	190
Data Files Generation	190
Immediate Threshold Refilling	195
<i>Execution Modes</i>	195
Online.....	195
Offline	198
Inverse	199
Best-Effort.....	200
Blank	200
Session-Based.....	200
Chargeable Items Acquisition Process	217
<i>Process Overview</i>	217
<i>Execution Modes</i>	218
Acquisition using the BART Server	218
Acquisition using the Core Server	218
Chargeable Items Consolidation Process	220
<i>Process Description</i>	220
<i>Consolidation Operations</i>	220
Tax Calculation Process	221
<i>Process Overview</i>	221
Inputs	222
Generated Tax Data	222
<i>Process Execution</i>	225
Tax Computing	225
Tax Adjusting and Applying	227
Taxation Modes.....	228
<i>Process Functions</i>	230
EU VAT with EU VAT Rules	230
EU VAT without VAT Rules	234
Flat Tax.....	237
U.S. Telco Tax (EZTax Tax System).....	240
No tax system.....	244
<i>Process Limitations</i>	245
Overrun with U.S. Telco Taxes (only relevant for the CC 2.0 and CC 3.0 model)	245
Session-based charging process with U.S. Telco Taxes.....	245
Operations on external references (only relevant for the subscriptions data model).....	245
Commissions or sponsorship (only relevant for the CC 2.0 and CC 3.0 model)	246
Credit Control Messages Conversion Process	246
<i>Process Description</i>	246
<i>AVP Dictionary</i>	246
<i>Service Dictionary</i>	247
<i>CCRs Mapping</i>	251
<i>CCAs Mapping</i>	251
Successful CCAs.....	252
Erroneous CCAs.....	252

Custom AVPs in CCAs	253
Process Limitations.....	254
8 PROCEDURES.....	255
Deploying SAP CC.....	255
Securing SAP CC.....	255
Starting SAP CC.....	255
Managing and Maintaining SAP CC.....	255
Managing Databases Partitioning	255
SAP ASE databases partitioning.....	255
Oracle databases partitioning	256
MS SQL Server databases partitioning	257
Synchronizing BIT Class, BIT type and Subprocess	258
Mechanism description	258
Mapping between CIT classes and BIT classes.....	259
Example.....	260
Mechanism Limitations	261
Synchronizing Chargeable Item Classes and Consumption Item Classes	261
Mechanism description	262
Managing consumption item classes	262
Creating consumption item mappings	263
Synchronizing Currencies.....	264
Monitoring SAP CC.....	265
Auditing SAP CC	265
Troubleshooting SAP CC	266
Upgrading SAP CC	266
TYPOGRAPHIC CONVENTIONS	267
ICONS.....	267
COPYRIGHTS	268
Disclaimer	269

1 Before You Start

Target Audience

- Solution and technology consultants
- Implementation project teams
- Application and system administrators
- Operation team experts
- Support specialists



NOTES

- Target audience should have a previous understanding and/or experience of client/server technologies in order to take benefit of this document
- This document is not included as part of the Installation Guides, Security Guides, Operations Guides or Upgrade Guides. Such guides are only relevant for a certain phase of the software life cycle, whereas the Application Help provides information that is relevant for all life cycle phases

Fundamental User Assistance and Guides

For a complete list of the available guides, primary help systems or other documentation related to SAP Convergent Charging 5.0, see SAP Help Portal at the following location: ► <http://help.sap.com/cc50> .



NOTE

SAP CC is part of the SAP solution *SAP Billing* based on SAP Business Suite or SAP S/4 HANA. Consult the central solution information at the following address: <https://wiki.hybris.com/display/ybilling/Library>

Additional Information

For more information about specific topics, see the quick links as shown in the table below:

Content	Quick Link
Security guides	http://service.sap.com/securityguide
Related SAP notes	http://support.sap.com/notes
Related release notes	http://service.sap.com/releasenotes
Released platforms	http://service.sap.com/platforms
SAP Solution Manager	http://support.sap.com/solutionmanager
How to configure SAP CC for CTS+	http://service.sap.com/sap/support/notes/2156128
Dedicated database for session-based charging data	http://service.sap.com/sap/support/notes/2189251

2 Getting Started

About this Document

The Application Help documentation describes the functional and technical aspects of SAP Convergent Charging 5.0, including descriptions of:

- The provided functional modules
- The implemented processes assigned to the functional modules
- The global architecture of the component

The Application Help contains the following main sections:

- *Overview*

This section provides a description of SAP CC, including a list of provided features or technical mechanisms, and a synthetic schema containing the different technical components which may play a role during the execution of the features.

- *Features*

This section contains the description of the different features provided by SAP CC.

- *Architecture*

This section provides an overview of the global architecture of SAP CC, with a description of the different technical components, including their provided tools used to configure, start, stop and manage them.

- *Master Data*

This section presents in a very synthetic way the main concepts managed by SAP CC.

- *Processes and Functions*

This section describes the different business processes and/or functions implemented into the SAP CC, including their possible execution modes.

- *Procedures*

This section describes different procedures related to SAP CC, from deployment recommendations to troubleshooting possibilities.

What's New in This Document

What's New in SP 4

The SP 4 version of this documentation contains the following modifications:

- New [Network Data Transfer](#) mechanism, used to transfer cached structures between instances
- Modification of the [Partitions mapping reorganization](#) and [Cache warm-up](#) sections to take into account:
 - The relation with the Network Data Transfer mechanism that can be used during smart warm-up operations
 - The System Maintenance mode feature that gives the possibility to switch the raters and guiders of your Core Server system to a new standby mode

What's New in SP 3

The SP 3 version of this documentation contains the following modifications:

- Enhancement of the schema containing the synthetic overview of the architecture, to display the communication channels used by the SAP CC Cockpit user interface and provide more visibility to the Session database
- New [Table rollover for rating sessions](#) feature, used to reduce the fragmentation of the Session database
- Enhancement of the [Databases](#) section, to describe the new Cockpit embedded database
- Some new apps are available in the SAP CC Cockpit user interface:
 - "Manage Mapping Tables", that you can use to search for, display, create, import, export and delete the mapping tables for your service providers and their end-customer services
 - "Manage Mapping Table Classes", that you can use to create, change or display some mapping table classes from the pricing catalogs for the end-customer services

What's New in SP 2

The SP 2 version of this documentation contains the following modifications:

- Some new apps are available in the SAP CC Cockpit user interface:
 - New "Display System Status" app that you can use to verify the status and availability of an SAP CC Core Server system in your SAP system landscape
 - New "Display Usage Metrics" app that you can use to:
 - Verify the SAP System Measurement based on usage metrics
 - Download a license auditing file as a GLAS report

What's New in SP 1

The SP 1 version of this documentation contains the following modifications:

- New "Item Files Monitor" application available in the SAP CC Cockpit user interface, the modern Web-based application to administer and supervise the SAP CC Core Server system

Document Abbreviations

The table below shows the list of abbreviations used throughout this document:

Abbreviation	Meaning
3GPP	3rd Generation Partnership Project
AAA	Authentication, Authorization and Accounting
AVP	Attribute-Value Pair
API	Application Programming Interface
BART	Batch Acquisition and Rating Toolset
BIT	Billable Item
BSS	Business Support System
CCA	Credit Control Answer
CCR	Credit Control Request

CDR	Call Detail Record or more generally Consumption Detail Record
CI	Chargeable Item
CIF	Charging output Integration Framework
CIT	Charged Item
CRM	Customer Relationship Management
CSR	Customer Service Representative
CSV	Comma Separated Value
CTS	Change and Transport System
DSR	Distributed Statistics Record
ECUR	Event Charging with Unit Reservation
GLAS	SAP Global License Auditing Service
GUI	Graphical User Interface
HA	High Availability
HCI	HTTP Communication Interface
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secured
IEC	Import Export Connector
IEC (Diameter)	Immediate Event Charging
IMSI	International Mobile Subscriber Identity
ISDN	Integrated Services Digital Network
LRU	Least Recently Used
MMS	Multimedia Messaging Service
MSISDN	Mobile Station ISDN Number
OCS	Online Charging System
OFCS	Offline Charging System
OSS	Operational Support System
QoS	Quality of Service
RAC	Real Application Cluster
RDBMS	Relational Database Management System
RIF	Rerating Integration Framework
RRL	Reservation Renewal Listener
SLD	SAP Landscape Directory
SCP	Service Control Point
SCUR	Session Charging with Unit Reservation
SMD	SAP Manager Diagnostic
SSL	Secure Sockets Layer
SSRL	Spending Status Report Listener

TAF	Transparent Application Failover
TIF	Transaction Integration Framework
TLS	Transport Layer Security
TV	Tag;value
URL	Uniform Resource Locator
USID	User Service Identifier
UTI	User Technical Identifier
VAT	Value Add Taxes
VSA	Vendor Specific Attribute
XML	eXtended Markup Language

3 Overview

Presentation

SAP Convergent Charging 5.0 is a modular component written entirely in Java. It is designed for the pricing, rating and charging of your services. Its modular architecture makes it easy to integrate and adapt in an existing environment. SAP Convergent Charging provides open APIs to connect to:

- Any customer management or provisioning software (CRM, self-care, and so on)
- Usage event producers (mediation, soft switches, SCP, and so on)
- Financial systems (legacy billing systems, ERPs, General Ledgers)
- Data warehouse systems
- And so on

SAP Convergent Charging provides multiple technical and functional features, which have been designed to fit the common using scenarios. It provides the following main functional features:

- Product modeling, used to create the master data related to the service provider, which are required when building your business environment
- Order management, which gives the possibility to create the master data related to customers
- Allowance management, which gives the possibility to manage the service add-ons for your charge plans and refill plans by modeling allowances and their associated usage
- Chargeable items acquisition, which gives the possibility to collect incoming events before executing further charging operations
- Online charging, which consists in charging in real-time the incoming events (with possible credit-control checking), determining the target account(s) which must be charged, and then generate data files containing all the data required by third-party systems for billing purposes (possibly aggregated when using the Charged Items Aggregation feature)
- Offline charging, which consists in rating at a deferred time the incoming events (related to postpaid accounts only, and thus without any possible credit-control checking), using a single or a batch mode
- Deferred charging, which consists in charging chargeable items which have previously been acquired and charged (using a blank mode) and which are sent again by SAP Convergent Invoicing for final billing purposes
- Manual capturing, which consists in charging a single chargeable item created and sent by SAP Convergent Invoicing
- Recharging, which gives the possibility to cancel previous charging operations and perform new ones on sets of chargeable items (which may have been previously billed) related to:
 - Subscriptions or provider contracts when recharging operations are initiated by the BART Server component
 - Provider contracts only when recharging operations are initiated by third-party systems such as SAP CRM or SAP CI, which only deal with master data related to the SAP CC 5.0 model
- Real-time credit-control, which gives the possibility to interact in real-time with a subscriber account to notify users and/or monitor its prepaid (or credit-limit) balance to ensure that a usage can be charged
- Prepaid accounts refilling, which gives the possibility to credit an account from a CRM application using manual or automatic (recurring or event-based) refill operations
- Taxes calculation, which gives the possibility to compute tax data on every incoming event according to the configured invoicing tax systems: EU VAT and/or U.S. Telco taxes (EZTax) or flat tax system
- Business notifications and business monitoring (spending status, account alerts) to manage in real-time the service consumption

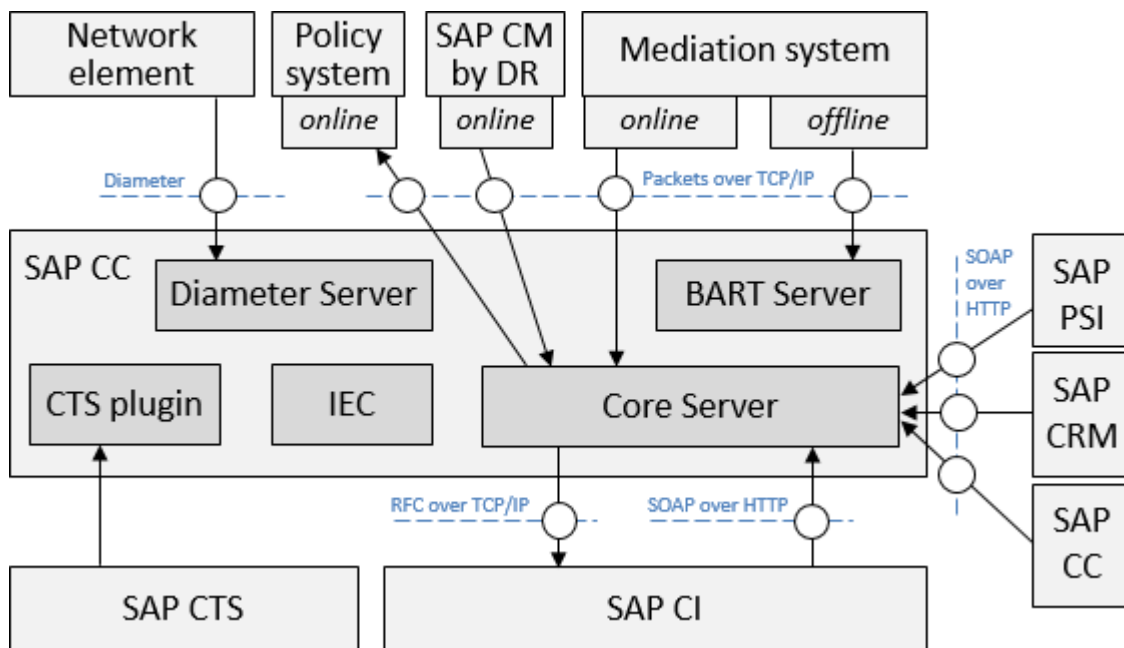
- Data files bulk loading, which gives the possibility to transfer the content of the generated data files from SAP CC to a third-party system, for various purposes such as billing, invoicing, reporting, business analytics, and so on

In addition, SAP CC provides the following main technical features:

- Transport of catalog data from an SAP CC system to one or several other SAP CC system(s)
- Transfer of provider contracts from an SAP CC system to another SAP CC system
- High availability, which relies on an architecture made up with multiple technical components dedicated to specific operations and whose deployment strategy avoids any service interruption
- High volume management, which ensures the scalability of SAP CC in case of increasing traffic or operations performed on large quantities of data (such as batch operations)
- System and data auditing, which provides detailed information related to the SAP CC product usage measurement and to the tracking of master data (and related user operations) modifications
- Data purging and archiving, which is used to remove from the back-end database some obsolete data, and thus ensure the global performance of the system
- Alerts and notifications framework, which is used for both business and administration purposes and gives the possibility to generate messages related to a usage of a marketable service
- User sessions management, which is used to control and/or monitor the connections to the Core Tool, BART Tool, and Cockpit user interfaces
- User isolation, which gives the possibility to specify whether service users must be isolated from individual users, or not, and thus control which kind of user can access user interfaces and/or execute HCI or Web Services operations
- Enhanced error framework, which concerns the implementation phase and gives the possibility to inform developers about the possible malfunctioning operations
- Enhanced logging and tracing frameworks, which ease both monitoring and troubleshooting operations by providing:
 - A complete reference of messages related to the execution of a given operation
 - Error files dedicated to the management of errors when dealing with data files generated by Raters and then bulk loaded into third-party systems by Bulkloaders
- Thread dump mechanism, which gives the possibility to generate dump files that contain information about the execution context of the different threads that are executed by the instances of the Core Server system, and that can be used for troubleshooting operations
- Enhanced integration interfaces, which provides different mechanisms such as web services, Java APIs or data files dedicated to the integration with third-party systems

All these features give you the possibility to ease your business and ensure its availability in case of increasing managed volumes. They have been implemented throughout the following technical components:

- Core Server, which represents the rating engine of the solution
- BART Server, a rating injector dedicated to batch rating and charging operations
- Diameter Server, used to translate incoming online CCR messages sent by a network element into SAP CC comprehensible events
- Core and BART databases, which are used to store all the data related to your business
- Other optional component: IEC, used to consolidate and schedule data transfers between SAP CC and third-party systems



The following table provides a list of dependencies between the functional features and the concerned technical components:

Feature	Core Server	BART Server	Diameter Server	IEC	Mediation System	CRM System	Billing System	CTS System
Product modeling	■					■	■	
Order management	■					■		
Allowance management	■					■	■	
Chargeable items acquisition	■	□		□	□			
Online charging	■		□		■			
Online charging with credit-control	■		■		□			
Offline charging	■	□		□	□			
Deferred charging	■						■	
Manual capturing	■						■	
Data files bulk loading	■						■	
Chargeable items rerating	■	□					□	
Prepaid accounts refilling	■				□	□	□	
EU VAT taxes calculation	■							
U.S. Telco taxes calculation	■							
Charged items aggregation	■						□	
Catalog Transport	■						□	□

- Mandatory
- Mandatory in a system landscape with an SAP ERP/FI-CA with an integrated SAP Solution scenario
- Optional

Technical System Landscape

The implementation of SAP Convergent Charging requires some mandatory components whose deployment depends on your business scenario.

For further information about the components availability and requirements, refer to the [SAP Convergent Charging Configuration Guide](#) and [SAP Convergent Charging Master Guide](#) documentations.

4 Features

Product Modeling

The Product Modeling feature gives the possibility to design all the master data related to Service Provider, in order to build your business.

For technical information about this feature, refer to both [Product Modeling Process](#) and [Order Management Process](#) sections.

Order Management

The Order Management feature gives the possibility to design all the master data related to end customers, in order to manage your subscribers and link this information to your business model.

For technical information about this feature, refer to both [Product Modeling Process](#) and [Order Management Process](#) sections.

Enterprise Credit Pooling

Enterprise Credit Pooling represents the ability to share counters (used to materialize minutes, SMSs, other credits) between multiple provider contracts in order to implement pooling plans designed for corporate subscriber accounts. For small business company, you can configure this model in SAP CC by sharing a pool of counters with provider contracts relating to the same subscriber account.

For more information, refer to the [adequate section](#) afterwards in this documentation.

Allowance Management

The Allowance Management feature gives the possibility to manage the service add-ons of your charge plans and refill plans by modeling allowances and their associated usage.

The Allowance Management feature gives the possibility to:

- Represent a right or a rule to consume one or several services within a period of time
- Dynamically add some consumption rights to your charge plans
- Report any allowance event by generating a transaction and its associated charged item according to a relevant mapping

Allowance Modeling

To build your business, it is necessary to model allowances by creating the following master data:

- [Allowance Interface](#), which contains the list of counters and parameters available for every allowance
- [Allowance Event Classes](#), which describe the allowances usage and their associated data
- [Allowance Logic](#), which describe the calculation algorithm of the Allowance Events
- [Allowance Plans](#), which are used to reference and customize your Allowance Logic

For each allowance logic, it is also necessary to describe the information which must be reported by:

- Selecting the charged item class to use

- Defining the mapping for the selected charged item class
- Referencing an aggregation policy

Allowance Usage Modeling

To model the usage of allowances, it is necessary to describe the interaction between allowances and charging or refilling logic (which corresponds to the life cycle of allowances) through algorithms. You configure the algorithms in the Price Plan, in the Refill Logic, or in the Allowance Logic master data and describe the following interactions:

- The creation of the allowance, which is based on a specified Allowance Plan
- The sending of the allowance events, which is based on a specified Allowance Event Class



NOTES

- Allowance Usage Modeling is based on specific allowance logic components and is therefore part of the Product Modeling Feature (for more information on allowance logic components, see the [SAP Convergent Charging Logic Components Reference](#) documentation)
- Allowances can be used within a subscriber account or can be used and shared across subscriber accounts (shared allowances)

Limitation

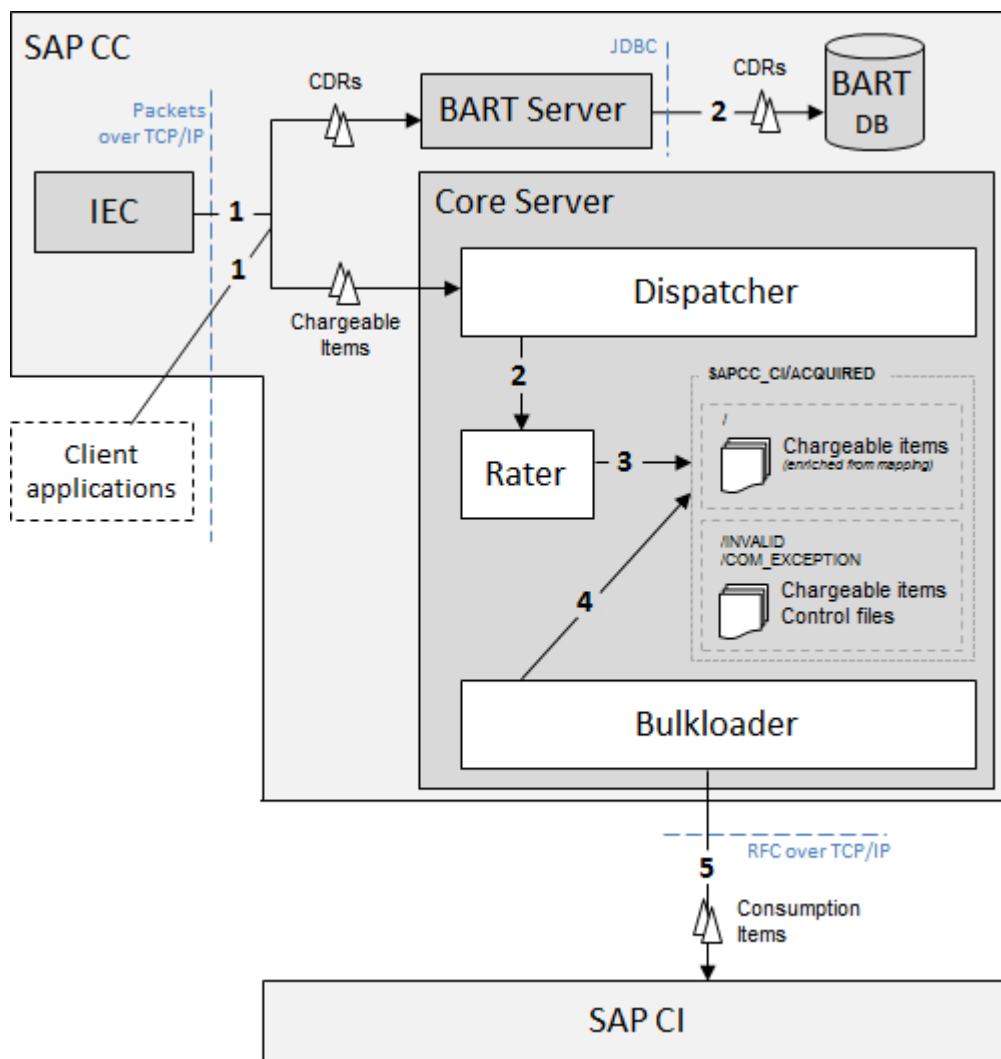
The Allowance Management feature is not compatible with the online stateless operations of the [Chargeable Items Charging](#) process.

Chargeable Items Acquisition

The Chargeable Items Acquisition feature gives the possibility to collect the incoming events synchronously sent by the IEC or by a third-party client application such as a mediation system (step 1) preliminary to further operations such as deferred charging, recharging, reporting, business analytics, and so on.

Acquisition operations themselves are performed by two components of SAP CC:

- The BART Server, using a synchronous execution mode and consisting in:
 - Consolidating each received chargeable item with complementary information retrieved from the Core Server
 - Validating these consolidated chargeable items
 - Recording these chargeable items into the BART database (step 2)
- The Core Server, using an asynchronous execution mode and consisting in:
 - Enriching each chargeable item with additional guiding information in order to execute the required charging operation (step 2)
 - Validating these enriched chargeable items and bulk writing them in data files (step 3), which will later on be taken into account by the [Data Files Bulk Loading](#) process (steps 4 and 5) to be recorded in SAP CI



For technical information about this feature, refer to the [Chargeable Items Acquisition Process](#) sections related to both [BART Server](#) and [Core Server](#) components, and to the [Chargeable Items Consolidation Process](#) section.

Online Charging

The Online Charging feature gives the possibility to process incoming chargeable items sent by the IEC or by a third-party client application such as a mediation system (step 1) in a connected mode (also called “online” mode). This feature represents the central feature of SAP CC, and consists in:

- Activating the charge which is associated to the external event (with a propagation to its possible dependent charges), in order to generate internal events
- Generating rating contexts to gather all the data which is necessary to perform the further steps
- Executing the price plans (step 2) in order to compute the amounts (prices, credits in counters, and so on) related to the different events located in the rating context
- Executing the charging plans (step 2) in order to update the determined balances with the computed amounts and calculate taxes

- Generating data files (step 3) containing all the information (charged items and possibly chargeable items) which might be necessary for billing and invoicing purposes, and handled by the [Data Files Bulk Loading](#) process (steps 4 and 5)

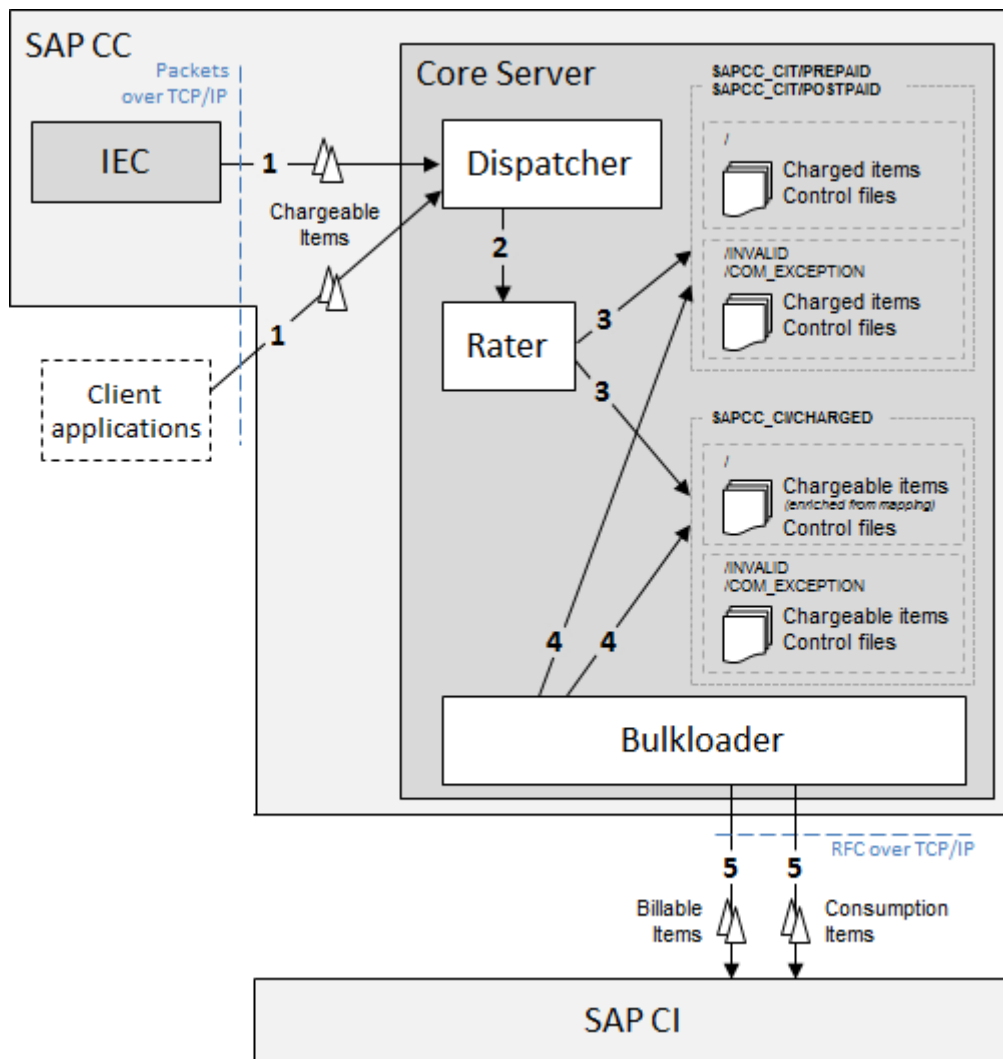
The Online Charging feature gives the possibility to execute in real-time the following operations:

- Event-based charging operations, which concern customer services characterized by a single event (such as SMS or MMS) which is sent before or after the service delivery. According to the needs, these charging operations can be executed in a single or batch mode, with possible credit-control and credit reservation features
- Session-based charging operations, which concern customer services characterized by a succession of single events and delivered over a given period, considered as a session of service consumption (such as phone calls, IPTV watching session, data download/upload, and so on) where the charged units are continuously consumed. This succession of single events represents a sequence of chargeable items which are charged, reserved or released in the related charging session. As these single events are associated to their order of arrival, **session-based charging operations cannot be executed in batch mode**
- Spending status monitoring, which gives the possibility to implement policy control based on information related to service usage, in order to take policy decisions accordingly

**NOTE**

To instantaneously adapt the service delivery and enhance the customer experience of the service provider, the session-based execution mode:

- Enables advanced credit control and reservation management (including reservation renewals) features
- Gives the possibility to execute session-based charging operations related to multiple customer services
- Supports refilling operations during the sessions



For technical information about this feature, refer to the [Chargeable Items Charging Process](#) section.

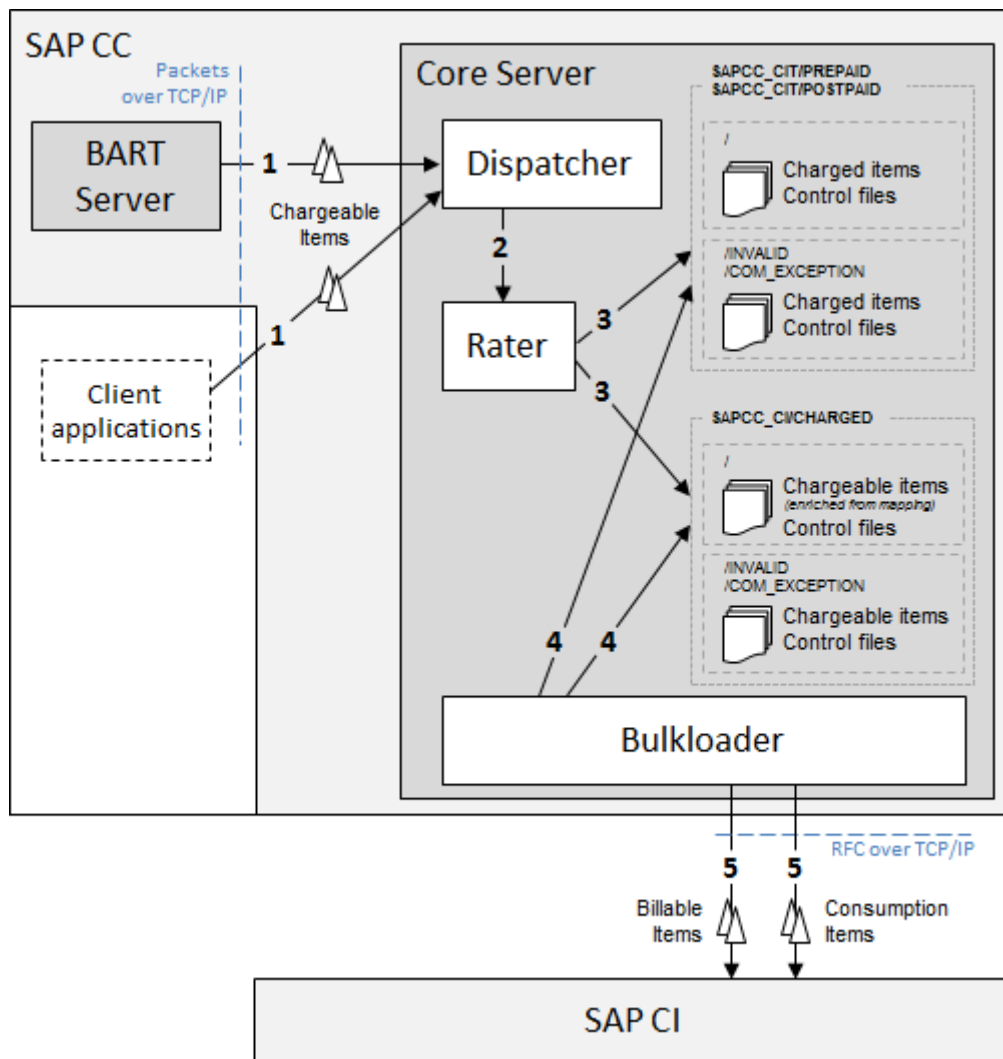
Offline Charging

The Offline Charging feature is similar to the Online Charging feature described above, except the fact that it works in a disconnected mode (also called 'offline' mode). This offline mode:

- Does not give the possibility to perform credit-control verifications
- Only gives the possibility to use postpaid accounts for your business, as prepaid accounts only work within connected systems

NOTE

To increase the performance of offline charging operations, it is also possible to use the BART Server, which gives the possibility to charge the offline events using a batch mode.



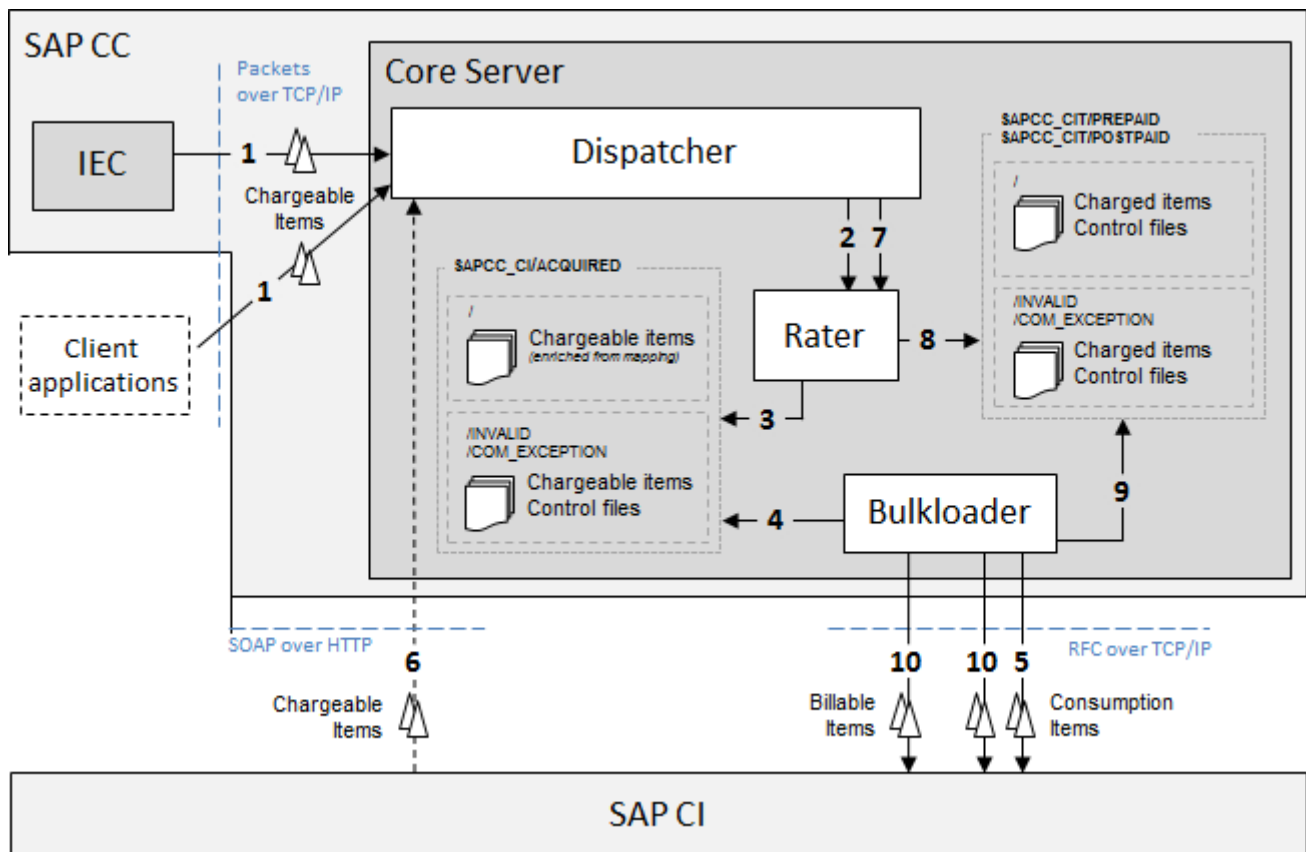
For technical information about this feature, refer to the [Chargeable Items Charging Process](#) section.

Deferred Charging

Some business configurations can lead to the acquisition of chargeable items (step 1) which voluntarily do not contain any information related to consumption prices (for example when the price of depends on a consumption volume which is computed at the end of each month).

When such a situation occurs, the acquired chargeable items must not be rejected and must be taken into account by the rating engine using a blank mode (step 2). They are flagged as uncharged chargeable items (step 3) and sent to SAP Convergent Invoicing (step 4), which:

- Records these uncharged chargeable items as consumption items (step 5)
- Enriches at a deferred time these items with price information required for charging purposes
- Re-injects at a deferred time (step 6) these chargeable items in SAP CC for final charging purposes (steps 7 to 10), using a dedicated web service

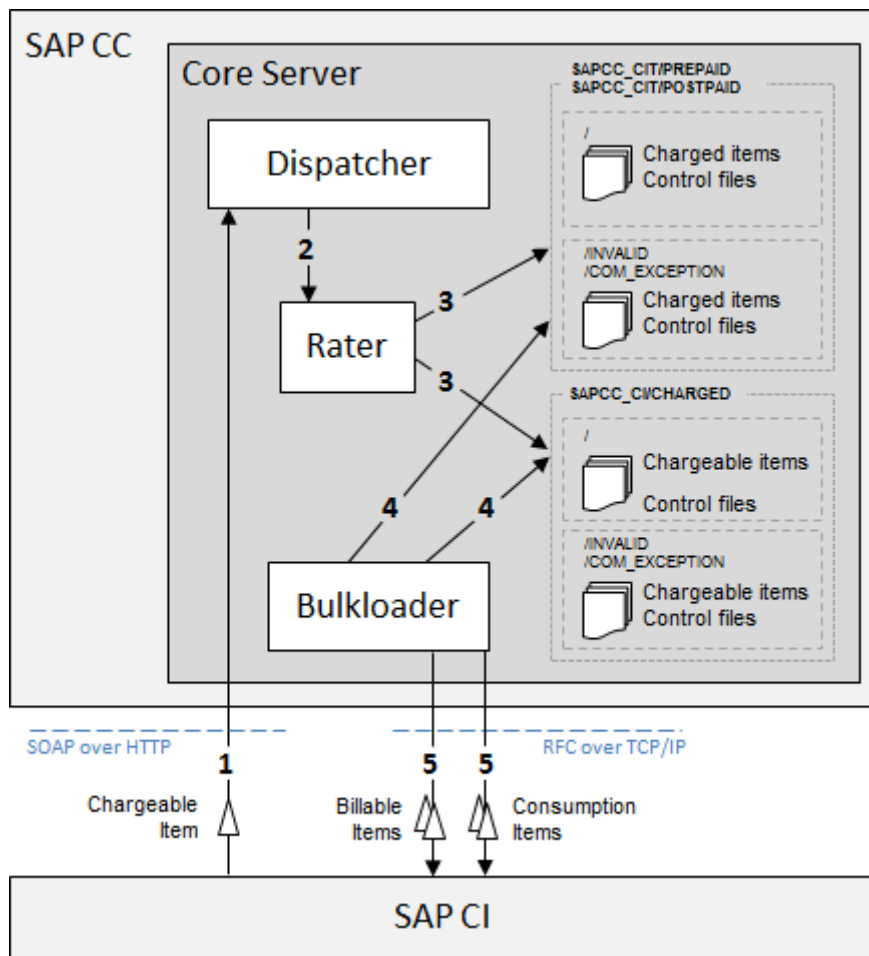


For technical information about this feature, refer to the description of the [Chargeable Items Acquisition](#) feature and to the [Chargeable Items Charging Process](#) section.

Manual Capturing

In an integrated *SAP Solution* scenario with the SAP ERP/FI-CA component of the SAP Business Suite, SAP Convergent Invoicing might sometimes need to correct a specific balance (in case of detected error, customer dispute and so on).

When such a situation occurs, SAP Convergent Invoicing needs to create a single chargeable item and send it to SAP CC (step 1) for charging (steps 2 and 3) and bulk loading (steps 4 and 5) purposes.



Data Files Bulk Loading

The Data Files Bulk Loading feature gives the possibility to transfer the content of data files generated by SAP Convergent Charging to third-party systems such as billing systems, invoicing systems, monitoring systems, and so on.

By default, SAP CC provides a set of processors which are configured to handle:

- Charged items, converted into billable items when sent to SAP CI
- Refill records, converted into billable items when sent to SAP CI
- Chargeable items, converted into consumption items used by SAP CI
- Business notifications about prepaid accounts, such as amount or expiration alerts, and expected by third-party systems such as SAP ERP or SAP CRM

These loading operations are performed by the Bulkloader instance, which periodically scans specific folders located on the file system, and containing the different data files generated during operations such as charging, refilling, recharging, and so on.

For technical information about this feature, refer to the [Data Files Bulk Loading Process](#) section.

Chargeable Items Rerating

The Chargeable Items Rerating feature gives the possibility to cancel previous charging operations and perform new ones on identified sets of chargeable items (which may have previously been billed). This feature is mainly used to correct errors such as:

- Incorrect or missing data in the incoming events
- Charging logic errors, which lead to incorrect computation operations during the charging process
- Failure during the execution of operations such as data conversion or bulk loading
- And so on

The billing system is able to cancel the original business transactions and to take into account the new business transactions (billable items in SAP Convergent Invoicing, charged items in other billing systems).

The execution of the Chargeable Items Rerating process depends on the elements which are concerned by the rerating operation:

- When the set of chargeable items to rerate contains elements referring to **offline** subscriptions, the rerating operations can only be driven by the BART Server system
- When only provider contracts are concerned, the rerating operation can be driven by:
 - SAP CC BART Server
 - SAP Convergent Invoicing with the consumption item management function enabled
 - A third-party application or system that implements the dedicated Web service of SAP CC and which gives the possibility to perform the different steps which are necessary to rerate a set of chargeable items



NOTES

- This Web service contains operations which have been implemented to fit an integrated *SAP Solution* scenario with the SAP ERP/FI-CA component of the SAP Business Suite, but it is also possible to use them in conjunction with another third-party system
- A procedure is available to migrate the execution of the rerating operations from the BART Server system to SAP CI in the SAP ERP system

For technical information about this feature, refer to the [Chargeable Items Rerating Process](#) section.

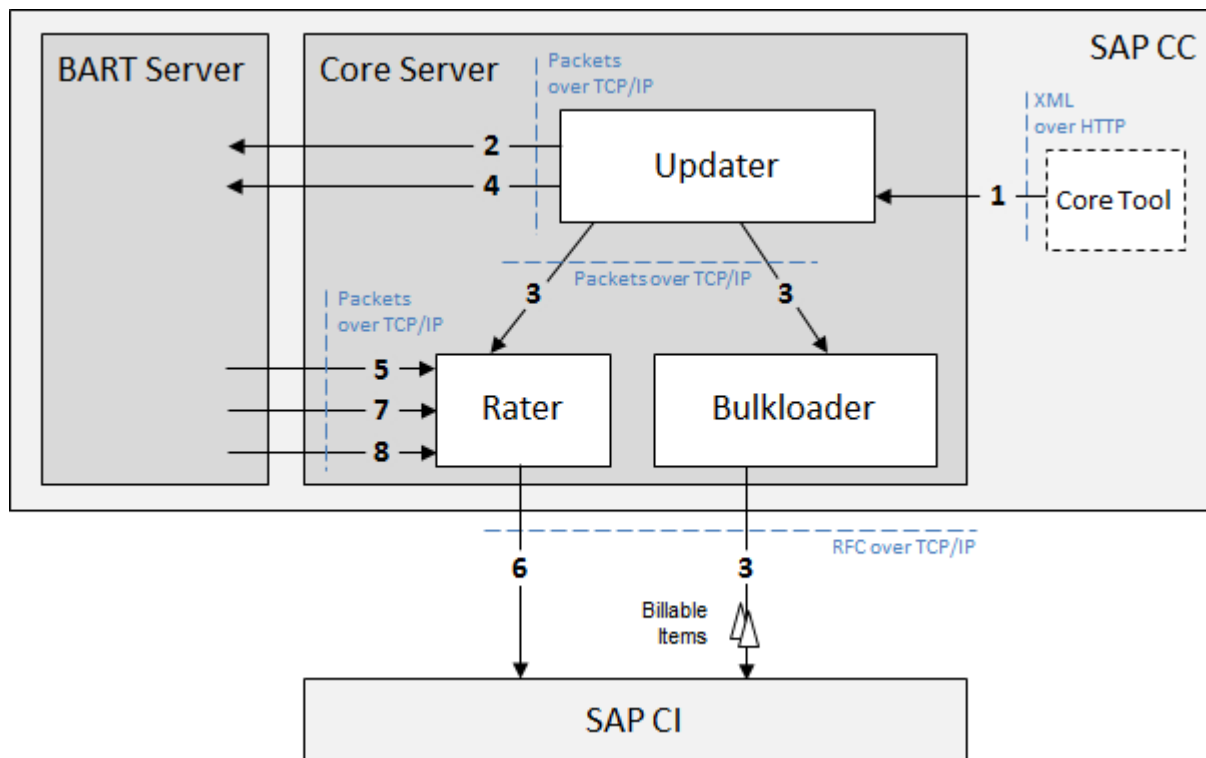
Rerating Operations driven by the BART Server

In an integrated *SAP Solution* scenario, the system landscape includes the following components:

- BART Server, which is responsible for the offline acquisition and manages the rerating operations
- SAP CC Core Server, which initiates the recharging operations related to the corrected chargeable items
- SAP CI, which manages the invoicing and billing operations

Rerating operations are initiated by the Core Tool user interface (step 1), and driven by the Updater which is responsible for:

- Opening rerating sessions (step 2) within the BART Server
- Preparing the rerating operations (step 3), in order to close data files and thus trigger the related bulk loading operations
- Starting the opened rerating sessions (step 4), within which the BART Server performs the rerating operations (steps 5 to 8)



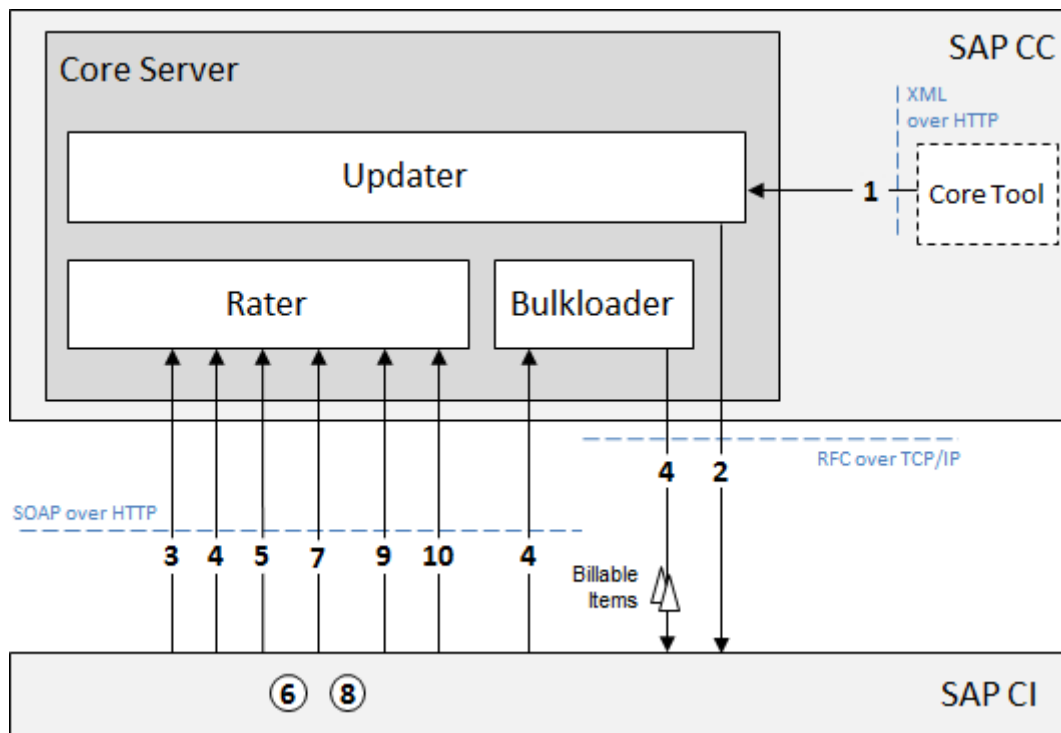
Rating Operations driven by SAP Convergent Invoicing

In an integrated *SAP Solution* scenario, the system landscape includes the following components:

- SAP CI, which is responsible for the final offline acquisition (data storage), for the management of the rating operations and for the management of the invoicing and billing operations
- SAP CC Core Server, which is responsible for the initial offline acquisition (batch acquisition, consolidation) and for the execution of the recharging operations related to the corrected chargeable items

Rating operations are initiated by the Core Tool user interface (step 1), within rating sessions opened by the Updater (step 2) within SAP CI which performs the following operations:

- Locking the dependent charging contracts (step 3)
- Closing the data files and thus triggering the related bulk loading operations (step 4) within SAP CC
- Retrieving the possible restoration points (step 5) from SAP CC
- Checking the BITs reversibility (step 6)
- Restoring the state of charging contracts (step 7) within SAP CC
- Reversing the BITs (step 8)
- Executing the new (batch) charging operations (step 9) within SAP CC
- Unlocking the recharged charging contracts (step 10) within SAP CC



Prepaid Accounts Refilling

Also known as a top-up process in the SAP Telecommunications industry, the Prepaid Accounts Refilling feature gives the possibility to credit the prepaid accounts of your customers. To refill their prepaid accounts, customers use points of services such as ATMs, call centers, or website, which all transfer credits from a refill card, a bank account or a prepaid account to SAP Convergent Charging in order to purchase goods and/or services.

Considering credits as monetary amounts (which may include a quantity of services, such as number of SMS, a quantity of data, and so on), SAP CC thus gives the possibility to use a given credit to refill a prepaid account:

- **Automatically:** SAP CC can automatically execute a refilling operation for a given customer, according to periodicity or when the prepaid account balance goes below a threshold defined by the customer
- **Manually:** When a customer requests a manual refill, the application that manages it (for example, a Voucher Management System) interacts with SAP CC and sends a refill request which includes some properties used to define the refilling operation (such as the origin of the operation, the associated gift, and so on)

For technical information about this feature, refer to the [Prepaid Accounts Refilling Process](#) section.

Tax Calculation

SAP Convergent Charging gives the possibility to calculate taxes by using:

- The “Value Added Tax” tax system, that is internally used by Raters to compute taxes for countries located into the European Union
- The “U.S. Telco Tax” tax system, that is used by Taxers when computing taxes for U.S. countries

Whatever the used tax system is, the tax calculation operations consist in:

- Calculating tax data from an updated rating context which contains tax settings of a given charge which is customized in a charge plan or in an offer and tax settings of the related subscriber account that holds an account to be charged
- Applying this tax data on charged transactions to:
 - Debit prepaid balances with the total amount including taxes
 - Enrich charged transactions with computed tax data to be taken into account by an external billing and invoicing system in order to calculate the final tax

The tax data is available in the charged items generated by SAP CC . In a solution scenario with the SAP ERP/FI-CA component of the SAP Business Suite, SAP CC converts the relevant tax data into billable items.

For technical information about this feature, refer to the [Tax Calculation Process](#) section.

Preconfigured EU VAT Rules

SAP Convergent Charging supports the following preconfigured VAT rules to configure the tax calculation:

The following preconfigured VAT rules are provided:

- EU VAT 2010 general rule for services
- EU VAT 2010 digital services
- EU VAT 2010 radio TV and telecommunication services
- EU VAT 2015 digital services
- EU VAT 2015 radio TV and telecommunication services

Preconfigured EU VAT Rule	Service Provider Zone	Place of taxation (B2B Services: Customer is a company)	Place of taxation (B2C Services: Customer is a private individual)
EU VAT 2010 general rule for services	European Union	Customer's place of establishment <i>The supply of services between businesses (B2B services) is taxed at the customer's place of establishment.</i>	Service provider's place of establishment <i>Services supplied to private individuals (B2C services) are taxed at the service provider's place of establishment (service supplier).</i>
	Third country	Customer's place of establishment	Service provider's place of establishment
EU VAT 2010 digital services	European Union	Customer's place of establishment <i>The EU VAT 2010 general rule for services applies</i>	Service provider's place of establishment <i>The EU VAT 2010 general rule for services applies</i>
	Third country	Customer's place of establishment <i>The EU VAT 2010 general rule for services applies</i>	Customer's place of residence <i>Electronically supplied services, provided by suppliers established in a third country to non-taxable persons established in the European Union, must be taxed at the place where the private customer resides.</i>
EU VAT 2010 radio TV and	European Union	Customer's place of establishment <i>The EU VAT 2010 general rule for services applies</i>	Service provider's place of establishment <i>The EU VAT 2010 general rule for services applies</i>

Preconfigured EU VAT Rule	Service Provider Zone	Place of taxation (B2B Services: Customer is a company)	Place of taxation (B2C Services: Customer is a private individual)
telecommunication services	Third country	Customer's place of establishment <i>The EU VAT 2010 general rule for services applies</i>	Place of effective service consumption <i>Radio and television broadcasting services and telecommunications services, supplied by suppliers established in a third country to non-taxable customers in the European Union, are taxable at the place where the private customer effectively uses and enjoys the service.</i>
EU VAT 2015 digital services	Any country	Customer's place of establishment	Customer's place of residence <i>Radio and television broadcasting services and telecommunications services, supplied by suppliers established inside or outside EU to non-taxable customers in the EU, are taxable at the place where the private customer resides.</i>
EU VAT 2015 radio TV and telecommunication services	Any country	Customer's place of establishment	Customer's place of residence <i>Electronically supplied services, provided by suppliers established inside or outside EU to non-taxable persons established in the EU, must be taxed at the place where the private customer resides.</i>

Charged Items Aggregation

The Charged Items Aggregation feature gives the possibility to merge multiple charged items into a single one (considered as a pending charged item), according to defined aggregation policies. These aggregation policies define some aggregation rules and reporting conditions which must be applied to charged items during charging operations, in order to determine whether 2 charged items must be aggregated or not, and decide whether a pending charged item must be reported or not. For further information about the aggregation policy master data, refer to the [Aggregation Policy description](#) located in the Master Data for the Service Provider section.



NOTES

- The Charged Items Aggregation feature is only available and relevant for charging operations executed in a session-based mode, and only concerns charged items related to consumed service usage (**i.e. related to the same customized charge or allowance logic**).
- When using allowances, charged items related to different allowances cannot be aggregated

To aggregate charged items, it is necessary to:

- Create and configure an aggregation policy
- Associate this aggregation policy (and optionally define a reporting scope) to the customized charges of the concerned charge plans or allowance plans



EXAMPLE

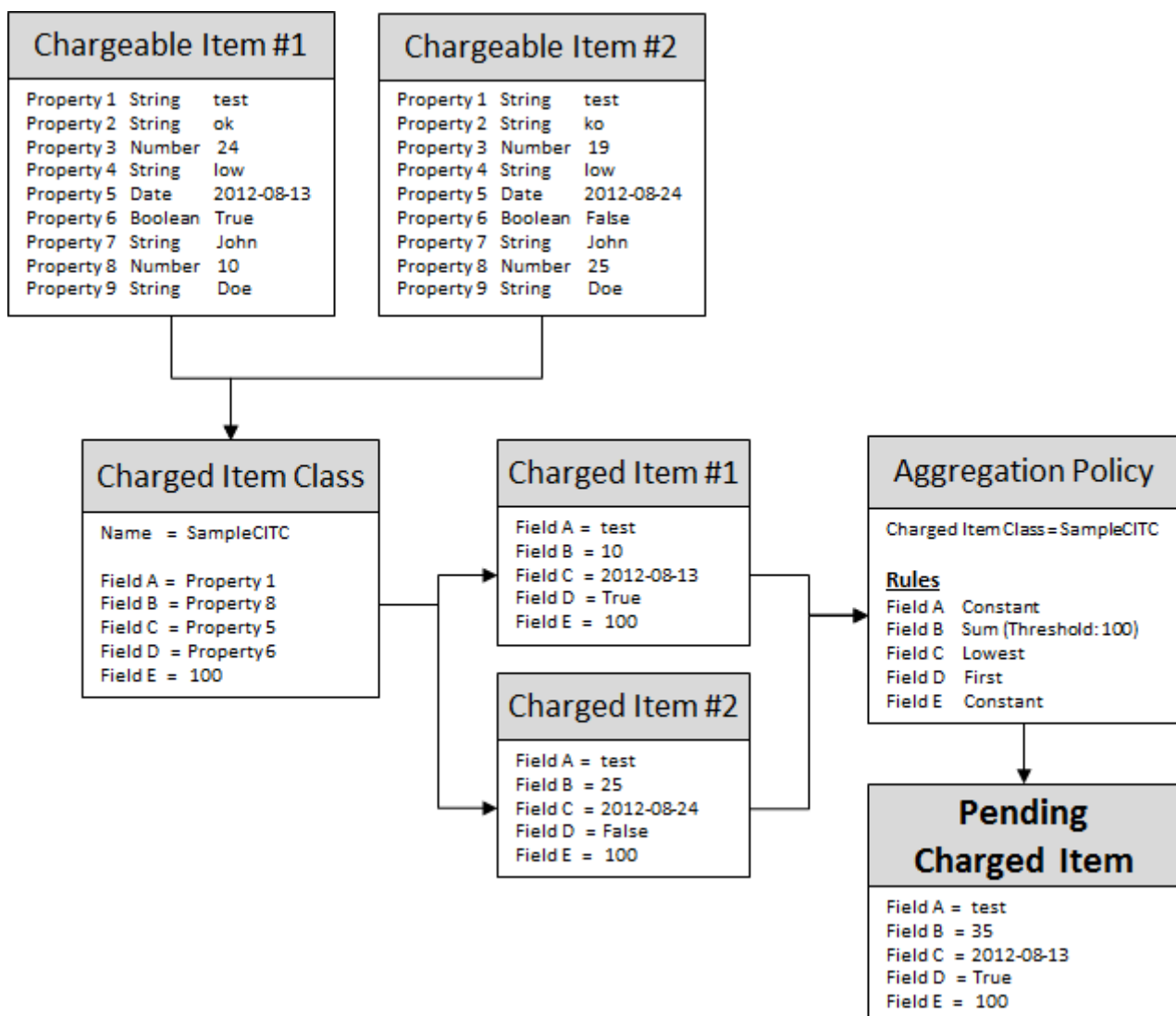
The following schema illustrates an overview of an aggregation of 2 charged items relating to the same consumed service and considering the use of:

- A charged item class named "SampleCITC", used to generate charged items containing 5 fields, from chargeable items containing 9 properties

- An aggregation policy based on the “SampleCITC” charged item class, defining aggregation rules for each field declared in the charged item class, but no specific aggregation period

Each chargeable item is processed and generates a charged item, according to the concerned charged item class. Then, the different rules defined in the aggregation policy are applied to the fields of these 2 charged items, in order to merge the charged items into a pending charged item, kept in the session for further aggregation operations.

Please note that the 2nd rule of the defined aggregation policy is associated to a threshold. But as the sum of the B fields of the charged items #1 and #2 does not exceed this threshold, the pending charged item does not need to be reported in the third-party billing system, and may thus be aggregated to further charged items.



Each aggregation rule relates to a property of the charged item class, and gives the possibility to compute the value of this property in the aggregated charged item, using the following functions:

- **Constant:** 2 charged items whose concerned property have the same value can be aggregated (keeping this constant value); otherwise, the pending charged item must be reported. This function is the default available function.

- **First:** The aggregated value corresponds to the not-null value of the first charged item
- **Last:** The aggregated value corresponds to the not-null value of the last charged item
- **Lowest:** The aggregated value corresponds to the lowest value of the pending and the current charged items. The null value is considered as positive infinity, except for Date properties where the null value is not taken into account.
- **Greatest:** The aggregated value corresponds to the highest value of the pending and the current charged items. The null value is considered as negative infinity, except for Date properties where the null value is not taken into account.
- **Sum:** The aggregated value corresponds to the sum of the values of the pending and the current charged items. The null value is considered as 0.



NOTES

- Each property of a charged item class is associated to a given type. The availability of the above described functions depends on the type of the properties:

Property Type	Constant	First	Last	Lowest	Greatest	Sum
String	■	■	■			
Number	■	■	■	■	■	■
Date	■	■	■	■	■	
Boolean	■	■	■			

- A threshold can be specified for each function associated to a Number property (except the Constant function). This threshold represents the value which leads to the reporting of the pending charged item every time it is exceeded
- Pending charged item(s) can also be reported in situations like:
 - A given property is associated to the Constant function, but the charged item to aggregate and the pending charged item have different values for this property
 - The charged item to aggregate and the pending charged item do not concern the same account type (and thus the same output channel). When such a situation occurs, the pending charged item is reported, and the charged item to aggregate becomes the pending one
 - The charged item to aggregate and the pending charged item thus do not have the same version of the aggregation policy due to modifications performed in parallel



CAUTION

The charged items classes contain properties which are fundamental regarding billing purposes (service identifier, account identifier, amount to be paid, and so on). The configuration of aggregation is not restricted, which means that it is possible to use any available function for any available property of the charged item class. It is thus highly important to pay attention when defining the aggregation rules of an aggregation policy, because some configurations can lead to inconsistencies regarding the global charging policy defined in price plans and/or charging plans. For example:

- Using a function different than **Constant** for the property associated to the account identifier leads to an aggregation of charged items related to different accounts, and thus prevents the dispatching of consumptions (and related amounts) between these different accounts
- Using a function different than **Sum** for the property associated to the amount to be paid prevents retrieving the sum of consumed service

- The amount (or tax amount) to be paid relating to an aggregated charged item can be different from the amount to be paid (or tax amount) resulting from the sum of every aggregated charged item, when a rounding mode is used in the pricing logic

The aggregation period represents the time during which the charged items used for **confirmation** purpose can be aggregated. The beginning of this period corresponds to the date of the first charged item used for **reservation** purpose. The end of this period depends on the activity of the related session:

- If a pending charged item has been reported due to an aggregation rule or due to session operation such as update or stop operations, then a new aggregation period is started using the date of the last charged item used for **reservation** purpose
- If no reporting is performed during the number of hours defining the maximum aggregation period, then the next charged item used for confirmation purpose will automatically report the pending charged item(s) and start a new aggregation period

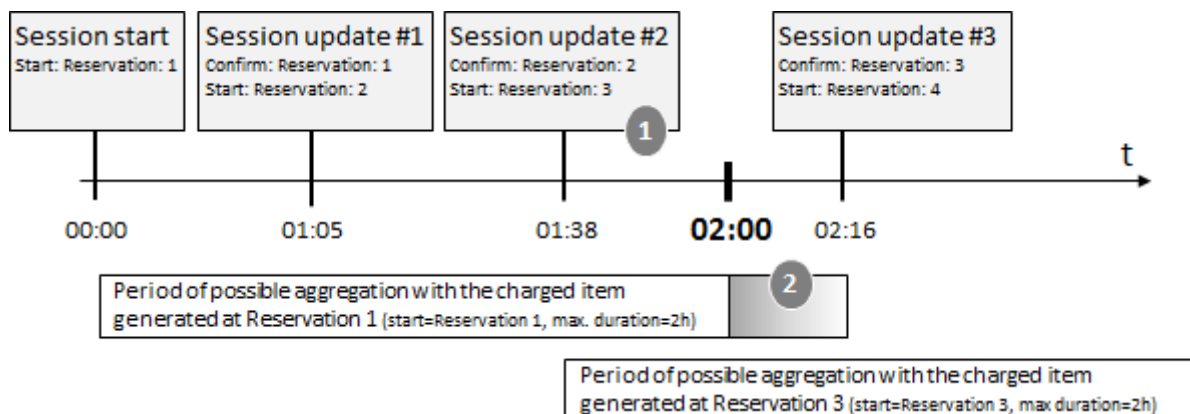
NOTE

The pending charged item(s) can also be automatically reported by the scheduler in charge of cleaning the ended sessions (due to session failure or “ttl” expiration for example). For further information about this scheduler, please refer to the [Session Cleaning operation of the Chargeable Items Charging Process](#).

EXAMPLES

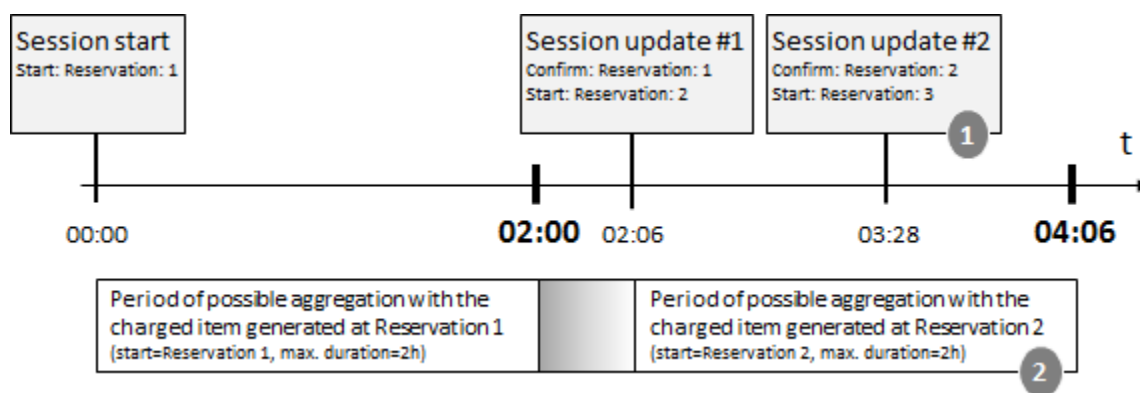
The following example illustrates the behavior of the Charged Items Aggregation feature during a session which is:

- Started at 00:00 with a first reservation of usage quota
- Updated at 01:05 with a chargeable item used to confirm the first reservation and reserve a second usage quota, converted into a charged item considered as the pending charged item
- Updated at 01:38 with a chargeable item used to confirm the second reservation and reserve a third usage quota, and converted into a charged item which is aggregated with the pending charged item (1) as the aggregation period is not ended
- Updated at 02:16 with a chargeable item used to confirm the third reservation and reserve a fourth usage quota, and converted into a charged item. This third session update occurs after the end of the aggregation period, which means that the pending charged item is ready to be reported. This reporting can either be done by the scheduler during the (2) extra time or by the update operation itself.



The following example illustrates the behavior of the Charged Items Aggregation feature during a session which is:

- Started at 00:00 with a first reservation of usage quota which represents the start of the period during which charged items can be aggregated
- Updated at 02:06 with a chargeable item used to confirm the first reservation and reserve a second usage quota, and converted into a charged item. As this session update arrives **after the end of the aggregation period**, it means that:
 - The charged item related to the first reservation is reported
 - The charged item related to the second reservation must be considered as the pending charged item, whose time corresponds to the start time of a new aggregation period (2)
- Updated at 03:28 with a chargeable item used to confirm the second reservation and reserve a third usage quota, and converted into a charged item which is aggregated with the pending charged item (1) as the aggregation period is not ended



A reporting scope can be configured within customized charges relating to master charges. This reporting scope specifies which charged items must be reported when a reporting condition is fulfilled:

- When using the “Charged Item” reporting scope, a reporting condition fulfilled for a charged item triggers the reporting of the related charged item only
- When using the “Session” reporting scope, a reporting condition fulfilled for a charged item triggers the reporting of all the charged items of the session

When a reporting condition is fulfilled for a charged item (either from a master charge, a dependent charge or an allowance), the “Session” reporting scope has the following behavior:

- If a constant aggregation rule is violated for a given charged item, all the pending charged items of the session are reported, and the new charged items related to the current charging operation are stored in the session.
- If a threshold is reached for a number value of a given charged item, all the pending items are aggregated with the new charged items related to the current charging operation, and then reported
- If the maximum aggregation period is reached for a given charged item, all the pending charged items are reported, and the new charged items related to the current charging operation are stored in the session (except if these charged items are themselves expired, which leads to their reporting)

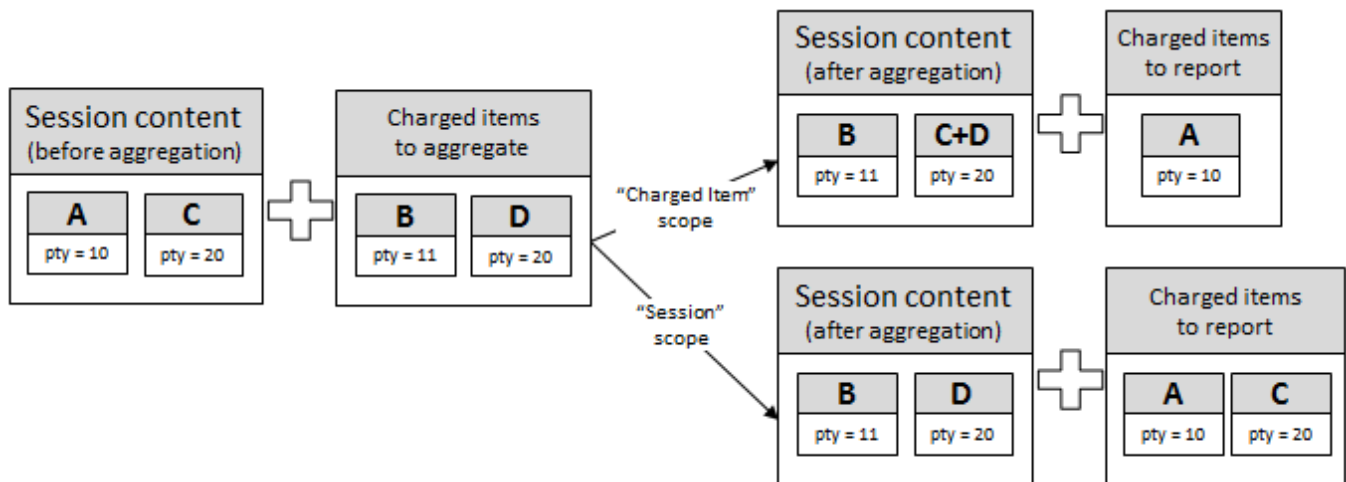
EXAMPLES

Considering the following elements:

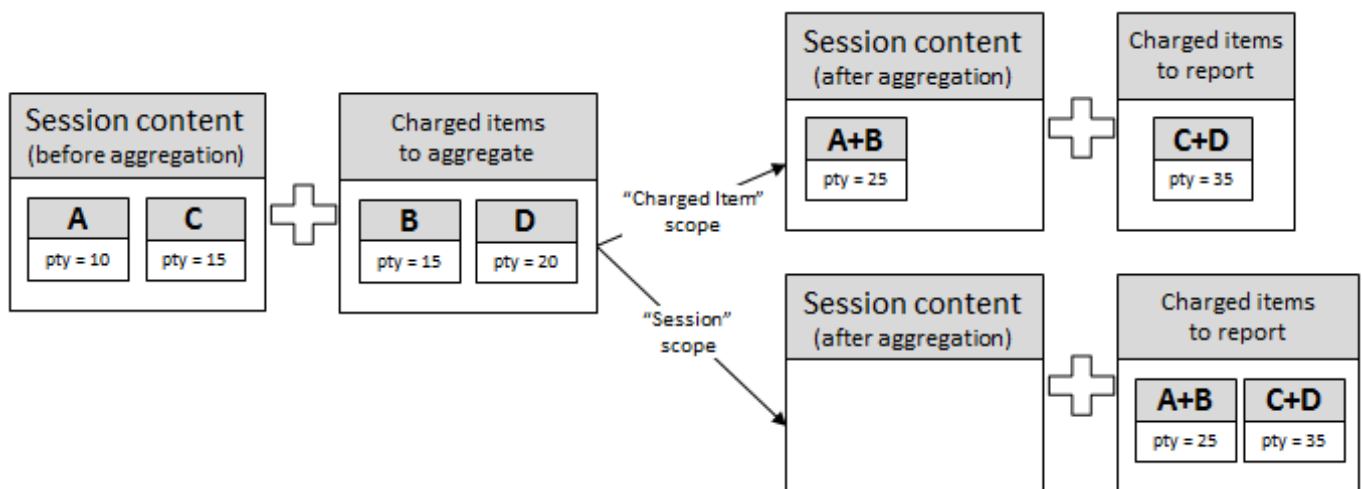
- A master charge MC
- A dependent charge DC triggered by MC

- 4 different charged items named A, B, C and D, each charged item containing a number property named “pty”:
- A and B relating to MC
- C and D relating to DC

The following schema illustrates the aggregation of the B and D charged items when using an aggregation rule for the “pty” property based on the “Constant” function:



The following schema illustrates the aggregation of the B and D charged items when using an aggregation rule for the “pty” property based on the “Sum” function and a 30 threshold:



Real-Time Credit-Control

Credit-control is a mechanism which directly interacts in real-time with an account and controls or monitors the charges related to a service usage. It globally consists in:

- Checking whether credit is available
- Reserving credit
- Deducting credit from an end user account when service is completed
- Refunding reserved credit that is not used

SAP Convergent Charging provides a real-time credit control mechanism which gives the possibility to:

- Control services and manage prepaid balances in real-time by checking the balances before delivering the concerned services to the customer
- Notify the customer when a balance threshold is reached
- Terminate the service session when the balance is depleted

This credit-control mechanism is based on messages exchanged between AAA clients (such as network elements) and servers. These messages can be:

- Credit-control requests (CCRs), sent by a client to a server
- Credit-control answers (CCAs), sent back by a server to a client which previously requested credit-control

Both CCRs and CCAs are handled by the Diameter Server component of SAP CC. This component is able to handle the following types of credit authorization models defined by the “Diameter Credit-Control Application networking” protocol:

- **Credit authorization with unit determination**, which corresponds to the following CCR types:
 - Event charging with unit reservation (ECUR), which consists in sending a request to authorize the customer service delivery and to reserve a service quota before delivering the service, followed by an additional request to confirm the finally used quota on service termination
 - Session charging with unit reservation (SCUR), which is similar to ECUR but includes intermediate reservation and confirmation requests before the final confirmation request on service termination
- **Credit authorization with direct debiting**, which corresponds to the Immediate Event Charging (IEC) CCR type, and consists in a single request to authorize the customer service delivery. The online charging system manages the user's account immediately after completing the credit authorization and debits monetary or non-monetary balances according to a rated amount. The network element delivers services only after receiving the authorization

 **NOTE**

The decision to use credit authorization with unit determination or credit authorization with direct debiting is based on the business requirements of the service provider. For further information about the IEC, ECUR, or SCUR CCR types, refer to the RFC 4006 documentation.

For more information about credit-control, refer to the description of the Diameter Server component or to the description of the [Credit Control Messages Conversion](#) process.

 **NOTES**

- SAP CC also provides a credit limit management feature, which gives the possibility to limit the usage of one or more services within a given period of time. When a defined limit is reached, the service is no longer accessible until the beginning of the next period. These credit limits are managed the same way as for prepaid balances, which means that SAP CC provides credit control for these limits
- SAP CC does not manage customer postpaid balances internally, but gives you the possibility to refer to external postpaid balances stored in a third-party billing system. During charging operations, SAP CC generates charged items or refill records, and then forwards them to a billing system, such as SAP Convergent Invoicing where the customer invoice is created and managed

Spending Status Monitoring

SAP Convergent Charging 5.0 provides a Spending Status Monitoring feature which enables a downstream system (policy, monitoring) to make real-time policy decisions based on customized information reported by the SAP CC system. This business information relates to both customer service usage made by an end customer and predefined spending thresholds. SAP CC converts and filters some counter values of provider contracts as form of spending statuses, and manages the reporting of these text indicators when they change.

In an online environment, you implement this feature to benefit from customer experience and predictive policy:

- The policy system manages the end customer monitoring and selects the monitored spending statuses
- The CRM or provisioning system takes into account some monitoring settings in pricing catalogs of a service provider to model commercial products and to manage provider contracts that include individual monitoring settings
- The downstream system and the SAP CC system must share the meaning of the defined spending statuses and their possible label values even there is no configuration synchronization
- The downstream system and the CRM or provisioning system must share some identification mechanisms for the end customers to be monitored

The Spending Status Monitoring feature takes part into the following processes:

- [Activation](#)
- [Chargeable Items Charging](#) (online, best-effort and session-based execution modes)
- [Prepaid Accounts Refilling](#)



EXAMPLE

For Telecommunications industry, you implement this function in your system landscape to provide a **Sy reference point** (in 3GPP architecture), which:

- Is located between the Policy and Charging Rules Function (PCRF) and the Online Charging System (OCS), to give the possibility to transfer subscriber spending information from the OCS to the PCRF. Status change of policy counters defined in OCS helps PCRF in making policy decisions based on these reported spending limits
- Supports the following functions to manage spending limit reports:
 - Request of policy counter status reporting from PCRF to OCS
 - Notification of policy counter status change from OCS to PCRF
 - Cancellation of policy counter status reporting from PCRF to OCS

Feature Configuration

The configuration of the Spending Status Monitoring feature is flexible and is available both in the pricing catalog of a service provider and in its provider contracts (customer master data):

- The Spending Status Monitoring feature adds monitoring plans in the pricing catalog which are associated to charge plans (or refill plans) and combined in a CRM application or provisioning system when modeling commercial products. The monitoring plans include different settings for the generation of spending statuses by identifying monitored counters in provider contracts, associated spending thresholds or limits (as form of particular range tables), and the identifiers of the spending statuses
- The monitoring configuration is finally set in the provider contracts. One or several contract items activate a monitoring plan and define the necessary technical data (end customer identification). The technical data knowledge must be shared between the CRM system and the policy system



NOTE

When SAP CC 5.0 is connected to SAP CRM 7.0 EhP3 (SAP enhancement package 3 for SAP CRM 7.0), SAP CRM cannot manage monitoring plan classes. You must enable a switch in SAP CC so that SAP CRM can combine monitoring plans as charge plans when modeling commercial products. For further information, refer to the [SAP Convergent Charging Configuration Guide](#) documentation.

Integration with a Policy or Monitoring System

The policy system registers to the SAP CC system and manages the required monitoring sessions for the end customers. It selects the reported spending statuses and takes policy decisions. It shares knowledge with the SAP CC system about the possible spending statuses and their labels.



NOTE

For further information about development information related to the communication between the SAP CC system and your policy or monitoring system (or another connector application), refer to the [SAP Convergent Charging Configuration Guide](#) and [SAP Convergent Charging Java Specifications](#) documentations.

Spending Status Reporting

SAP CC gives end customers the possibility to monitor and manage related spending status reports when relevant changes are detected. These reports are sent by SAP CC to the connected and registered policy or monitoring system in real time.



NOTE

Non-acknowledged reports are automatically resent by SAP CC.

Functions and Mechanism Details

To track changes, SAP Convergent Charging executes the following operations when receiving one of the available operation requests (listed above):

- For a provider contract which has been selected during online charging or refilling operations, the SAP CC system identifies if monitoring sessions have been requested by the policy system
- In the provider contract, the SAP CC system considers all the contract items which activate a monitoring plan. The system uses this monitoring plan to determine the list of all spending statuses to monitor and report
- The SAP CC system retrieves the label of each selected spending status in order to detect changes after the processing of the business operation. To retrieve the correct label, the system uses the dedicated range table identified in the monitoring plan and maps the counter value to the ranges that represent the spending thresholds. The output value of the range table represents the label of the corresponding spending status before the processing of the business operation which triggered the detection change function. If this execution of the range table leads to no output value, the system considers that the spending status is not defined at this time
- The SAP CC system executes the business operation (charging, refilling, activation)
- The SAP CC system computes again the label of the relevant spending statuses and thus detects the possible change. Changed statuses are grouped together.

- For each identified active monitoring which is concerned by a status change, the SAP CC system generates and sends a new spending status report to the downstream system via the Packets over TCP/IP communication channel



NOTES

- The offline charging operations do not influence the spending status reporting. SAP CC does not take into account some counter changes relating to batch charging operations available via the Packets over TCP/IP or the Web Services communication channels
- SAP CC does not consider the potential changes relating to the “Change charging contract state” operation of the “Contract State Management” Web service. This operation is not part of the online charging services. It is dedicated to initial mass provisioning or customer data migration

Limitations

- The Spending Status Monitoring feature is not compatible with the stateless operations of the [Chargeable Items Charging](#) process executed in online mode
- The Spending Status Monitoring feature does not support the Enterprise Credit Pooling feature (which consists in sharing a pool of counters between a parent provider contract and its linked provider contracts)

Catalog Transport

The Catalog Transport feature gives the possibility to transport data of a given catalog from an SAP CC system to one or several other SAP CC system(s). The data to be transported is included in a dedicated “[Change List](#)” master data, which can be sent to a list of transport destinations through transport requests:

- Handled by the “source” SAP CC system using a dedicated scheduler
- Handled by a SAP CTS system working in conjunction with a dedicated SAP CC CTS plugin

Transport Destinations

A Transport Destination corresponds to a remote SAP CC system to which change lists can be transported. Every transport destination must be configured, either within every “source” SAP CC system, or within the SAP CTS system.

For further information about the configuration of the list of transport destinations within a SAP CC system, refer to the “System Settings for Catalog Transport” section of the [SAP Convergent Charging Configuration Guide](#) documentation. For further information about the configuration of transport destinations within a SAP CTS system, refer to the “[How to configure SAP CC for CTS+](#)” documentation.

Transport Requests

Transport requests handled by the SAP CC system

SAP CC transport requests represent a structure which contains one change list and one reference of a transport destination on which the change list must be applied. These transport requests are created only when a user schedules or executes a given change list, for each concerned transport destination. Then, these transport requests are taken into account by a dedicated scheduler in charge of executing them:

- Immediately if the user decided to “execute” the change list

- At a deferred time if the user decided to "schedule" the execution of the change list



CAUTION

When the scheduler takes a transport request into account, it retrieves from the system all the information related to the concerned transport destination, and adds this information to the transport request. Due to the gap between the creation of the transport request and its execution by the scheduler, **the information related to the transport destinations might have changed** (parallel modification or deletion, for example).

Whatever the execution mode is, transport requests have a status which can change upon its lifecycle:

- **Created**, which represents the initial status informing that the transport request has been created but not yet taken into account by the dedicated scheduler in charge of executing it
- **Executed**, which means that the transport request has been taken into account by the scheduler (successfully or not)



NOTES

- For further information about the scheduler in charge of executing transport requests, refer to the [SAP Convergent Charging Configuration Guide](#) and [SAP Convergent Charging Parameters Reference](#) documentations
- It is only possible to unschedule transport requests which have not yet been taken into account by the scheduler. When a transport request is unscheduled, it is deleted from the system
- When considered as "executed", a transport request cannot be scheduled or executed again. It is only possible to create a new transport request (which can be similar to the executed one)
- As users can execute or schedule a given change list multiple times on a given transport destination, more than 1 transport request can refer to a given transport destination
- When multiple transport requests must be executed, the scheduler orders these transport requests as follows:
 - The associated change lists are retrieved, and ordered by their creation date. Transport requests associated to the oldest change lists are executed first
 - Within a given change list, the transport requests are ordered by their creation date
- To ensure a successful execution of a transport request, you must check that the configuration of the remote system(s) is the same as the configuration of the "source" system(s). This includes the following data
 - Currencies (in both SAP CC and SAP CI)
 - Public holidays (in SAP CC)
 - Default charged item mapping (in SAP CC)
 - Default chargeable item mapping (in SAP CC)
 - The maximum allocation memory for the Updater instances (in SAP CC)
 - Billable Item Classes (in SAP CI)
 - Consumption Item Classes (in SAP CI)



CAUTION

When a transport request is applied on a remote SAP CC system, the cached structures of its rating instances are automatically refreshed to immediately take into account the new version of the transported objects. As a consequence, it is highly recommended to be careful when transporting master data that are not managed chronologically.

The execution of a transport request is performed by the active updater of the "source" SAP CC system, and consists in the following operations:

- Computing the list of master data to transport, according to the content of the change list
- Converting this list of master data into compressed binary data
- Updating the status of the currently concerned destination from "Created" to "In Progress"
- Retrieving the instance map of the remote SAP CC system in order to get the address of the active updater

**NOTE**

If an error occurs while retrieving the instance map of the remote SAP CC system, the status of the transport destination is set to "Authentication Failure" or "Communication Failure", according to the situation.

- A "changeListApply" operation is performed on the "transport" Web service hosted by the active updater of the remote SAP CC system. The request contains the compressed binary data, and is sent asynchronously to give the "source" updater the possibility to manage the following transport destination without waiting for the answer of the remote updater

**NOTE**

If an error occurs while sending the request to the updater of the remote SAP CC system, the status of the transport destination is set to "Authentication Failure" or "Communication Failure", according to the situation.

On the remote SAP CC system, the execution of the "changeListApply" operation consists in the following operations:

- Retrieving the list of master data to handle, from the provided compressed binary data

**NOTE**

Depending on its size, it might be necessary to uncompress the received binary data in the temporary folder of the operating system of the remote SAP CC system. It is highly recommended to size this folder in order to ensure that the binary data can be correctly uncompressed.

- Creating catalog entries for each retrieved master data, if necessary
- Importing all the retrieved master data, including their full chronologies
- Refreshing the cached structures of the rater instance(s), to make the modifications immediately available
- Returning a response to the "source" updater, to inform about the status of the operation, which can either be successful or failed

**NOTE**

By default, responses are returned asynchronously. When communications between the SAP CC systems are secured, these response are synchronously returned.

When receiving the response, the "source" updater updates the status of the corresponding transport destination related to the transport request. If a successful response is received, the status of the destination is set to "Successful". Otherwise, this status is set to "Failed", with an additional failure message, an error stack and a failure category which can be one of the following:

- **Authentication Failure**, which can concern failures such as invalid password, wrong login, and so on

- **Communication Failure**, which can be due for example to a network issue preventing the connection to the remote SAP CC system
- **Invalid**, which means that a master data included in the change list is invalid and cannot be applied on the destination
- **Incompatible Configuration**, which means that the configuration of the remote SAP CC system is not compatible with a master data included in the change list
- **Illegal State** or **Temporary Illegal State**, which mean that the change has not been applied because of the state of the remote SAP CC system

Transport requests handled by the SAP CTS system

The SAP CTS system provides an ABAP Web Dynpro application named “Transport Organizer Web UI”, which gives the possibility to:

- Define and configure a landscape containing the different SAP CC systems considered as transport destinations
- Manage transport requests, from creation to sending and monitoring

Each transport request created within the Transport Organizer Web UI includes a SAP CC file which has been exported from the “source” SAP CC and which contains the change list to transport. For further information about the management of transport requests within the Transport Organizer Web UI, refer to the [“How to configure SAP CC for CTS+”](#) documentation.

Provider Contract Transfer

The Provider Contract Transfer feature enables you to transfer provider contracts from one SAP CC system to another one.

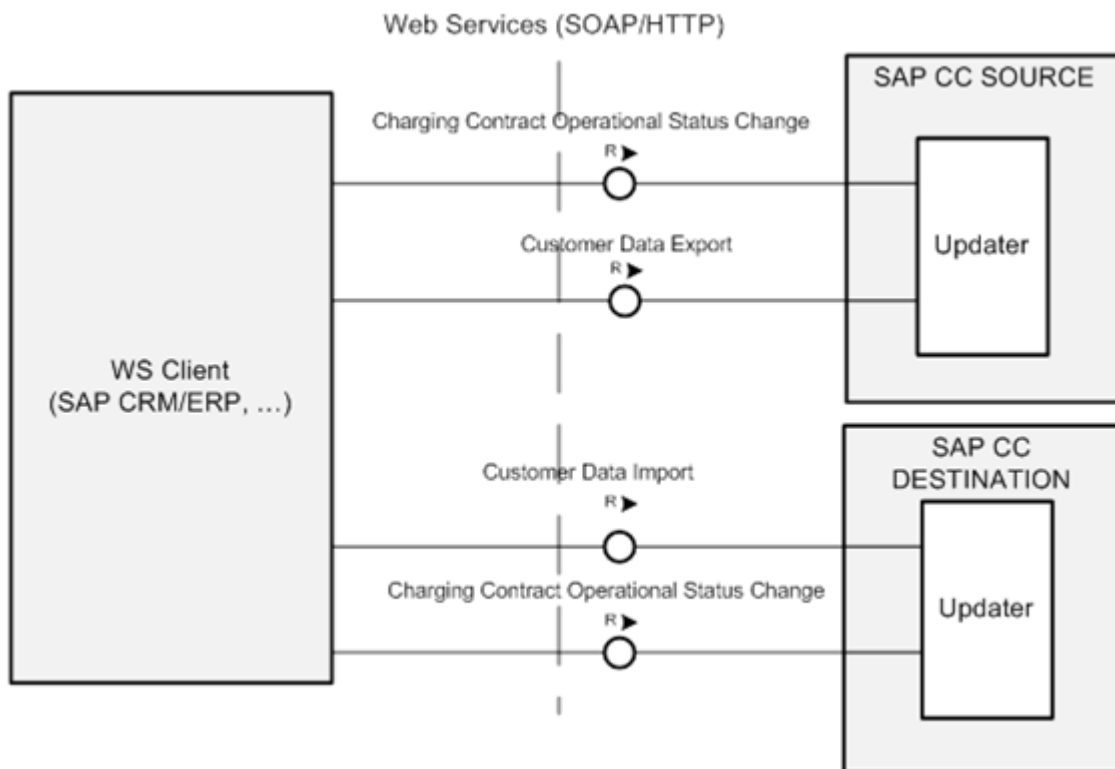


NOTE

In an integrated scenario with SAP ERP and SAP CRM, SAP CRM drives the contract transfer. See the [O2C Integration Guide](#) for more information on this topic.

Overview

SAP CC Web Services are enhanced to export and import customer data and to lock provider contracts during the transfer. By combining the new operations, you can easily move a provider contract or a subscriber account from one SAP CC system to another one.



Overview of the Contract Transfer feature

 **NOTE**

Before transferring provider contracts, make sure that the configuration of the destination system is the same as the configuration of the source. See [SAP Convergent Charging Web Services](#) documentation for details on the prerequisites to be fulfilled for Contract Transfer.

Customer Data

Customer data include:

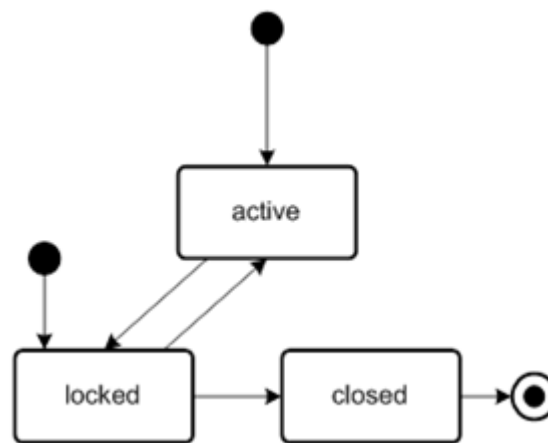
- Subscriber accounts, their prepaid and external accounts and the subscriber mapping and range tables
- Provider contracts, the contract states and the allowances

Operational Status of a Provider Contract

Provider contracts are enhanced with an operational status which enables to lock them during the transfer. The available values for the operational status are listed below:

- **Active:** The provider contract is fully operational.
- **Locked:** The provider contract is temporarily locked. SAP CC processes are not available for this contract.
- **Closed:** The provider contract is deactivated.

The following figure shows the available values for the operational status and how they link together:



Values of the operational status

Export and Import Operations

SAP Convergent Charging Web Services are enhanced with the following service operations:

- Customer Data Export: To export customer data from an SAP CC system to another
- Customer Data Import: To import customer data into an SAP CC system from another
- Charging Contract Operational Status Change: To lock a contract before transferring it from an SAP CC system to another one or to unlock a contract imported into the SAP CC destination system

NOTE

The new WS operations can be audited in Core Tool.

NOTE

You can rerate or recharge provider contracts only from a date later than the date when they were imported into the SAP CC destination system.

For more information about the Contract Transfer feature, refer to the [SAP Convergent Charging Web Services documentation](#).

High Availability

SAP Convergent Charging provides a high availability feature which relies on both hardware and software specifications and configurations. According to the needs, it is then possible to implement a strategy which avoids any service interruption in case of element failure (databases, server instances, platforms, and so on).

Hardware High Availability

As far as the hardware configuration is concerned, the HA feature mostly depends on the deployed landscape (hosts specifications and repartition, RDBMS version and strategy, and so on). For further information about these landscapes configurations, refer to the [SAP Convergent Charging Sizing Guidelines documentation](#).

Software High Availability

As far as the software configuration is concerned, the HA feature depends on the number and type of deployed instances:

- All deployed Dispatchers are contacted by client applications using a transparent round-robin mode. If a slave Dispatcher fails, another one will be automatically contacted. If the master Dispatcher fails, the first available slave Dispatcher becomes the master one. As a consequence, all the incoming requests are handled.
- If the active Updater fails, client applications contact the first available backup Updater (automatically started by the master Dispatcher)
- All deployed Raters and Guiders are contacted according to a partition identifier. If such a server instance fails, the master Dispatcher reorganizes its partition map among the other server instances, and contacts the newly concerned one
- All deployed Taxers are contacted using a transparent round-robin mode. If a Taxer fails, the next available one is contacted
- No HA is provided for the Bulkloaders. If a Bulkloader (assigned to a Rater) fails, bulk loading operations will not be performed until the failed instance is restarted

Heartbeat

The heartbeat mechanism gives the possibility to check if connections with a given instance are down or alive. A heartbeat message is sent on the messages over TCP/IP communication channel, without any expected response but only with an acknowledgement with a configurable timeout (set by default to 2 seconds). After 3 attempts without any response, the timeout is considered as expired and the concerned instance is then considered as down. The heartbeat mechanism continues (in background) to try to open a connection with the down instance.



NOTE

A dedicated API named `FoundLostRatingListener` and located in the `com... .cnd.message` package can be used to be informed when a connection is down.

Databases active-active high availability

The Core database of SAP CC has been designed to take benefit of the Oracle RAC (with partitioning option) and DB2 pureScale mechanisms which increase the global scalability of the solution by giving the possibility to:

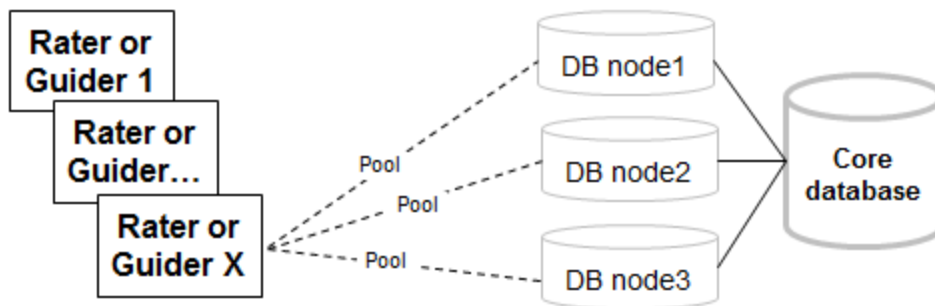
- Run multiple database instances on multiple nodes
- Share a single physical database between multiple database nodes
- Control and manage the same data and initialization files whatever the running database node is
- Have dedicated or shared resources (individual log files, shared data files, and so on) for each database node
- Simultaneously execute transactions on a single database from different server instances

When SAP CC data are partitioned, each Rater or Guider manages a set of business objects which is isolated into a dedicated database instance. Connecting each Rater or Guider to a dedicated database instance gives the possibility to:

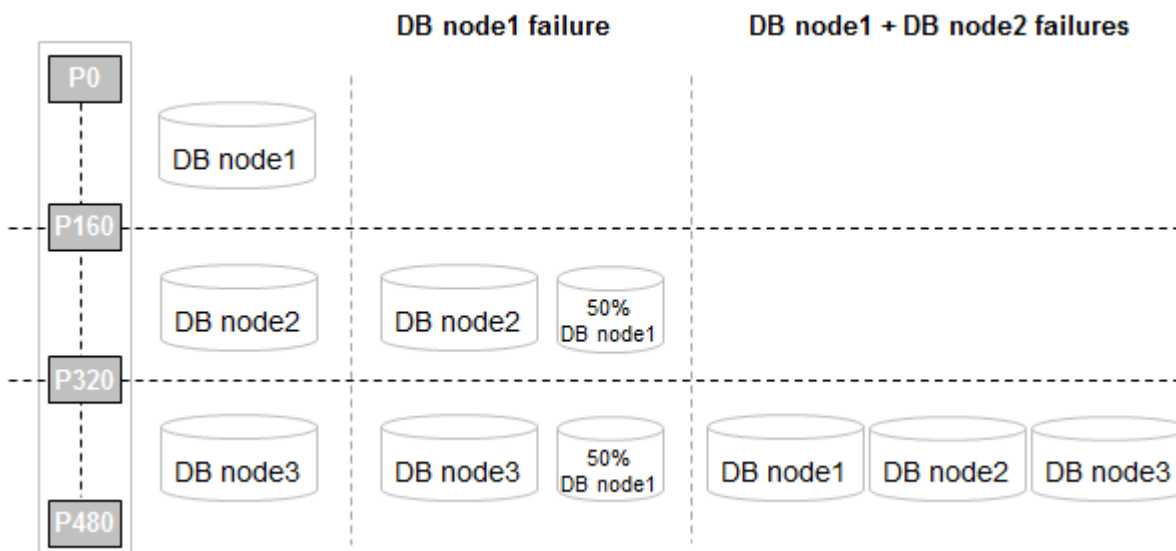
- Subdivide the interactions between the running server instances and thus provide load balancing and failover features (using an optimized mechanism mentioned afterwards)
- Ensure that database resources are isolated and dedicated to a given server instance

An optimized failover mechanism has been implemented in SAP CC to handle failures of the databases instances. This mechanism relies on the following technical parts:

- A proprietary mechanism which reallocates the partitions handled by a failed database instance to the other running ones
- Pools of database connections (handled by Raters and Guiders for each running database instance), which gives the possibility to immediately switch the connections from a failed database instance to an alive one
- A proprietary heartbeat mechanism, which gives the possibility to check whether a database instance is still alive or not, without any impact on the global performance of the solution



In case of database instance failure, the SAP CC system automatically switches the connections to another database instance in order to guarantee the global charging service, even if only one database instance is alive. Note that some incoming events might be lost in case of failover, depending of the database state when the failure occurs (the commit might be totally, partially or not validated). **It is thus highly recommended to re-emit the failed or timed-out events when a database instance failure occurs.**



NOTES

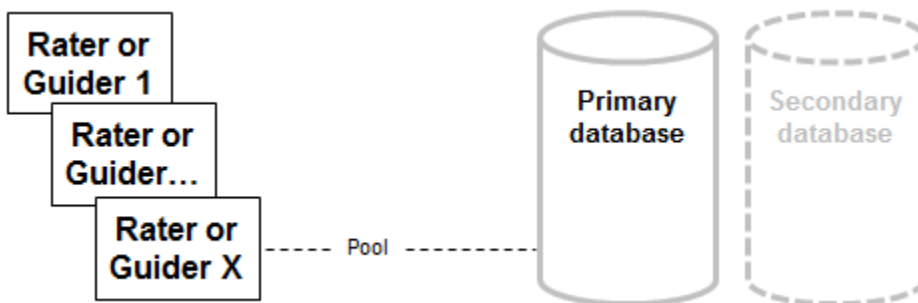
- A maximum number of 12 database nodes can be managed and allocated to the deployed Raters and Guiders. As the number of deployed Raters must be a divider of 480 and the number of deployed Guiders must be a divider of 240, this distribution can be uniform or uneven. To get a uniform allocation between Raters and Guiders, 2, 3, 4, 5, 6, 8, 10 or 12 database nodes must be configured. Using another configuration leads to an uneven allocation

- When the partitioning mechanisms are not implemented, refer to the description of the active-passive high availability located afterwards

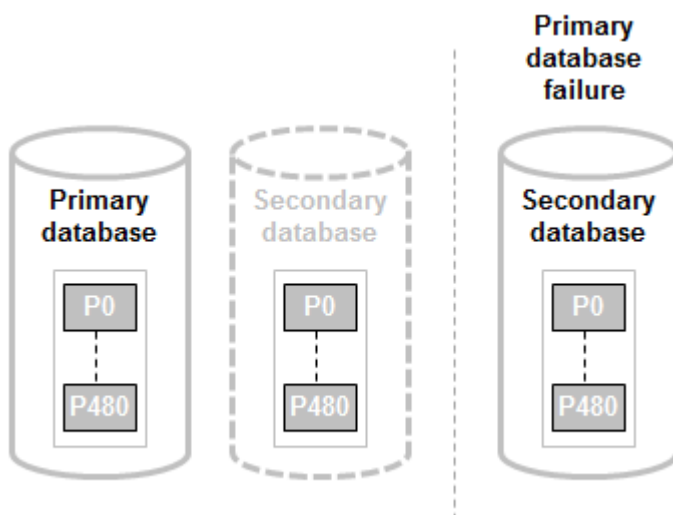
Databases active-passive high availability

The active-passive high availability mode implemented in SAP CC relies on the following elements:

- A single-instance database considered as the primary database
- Another single-instance database considered as the secondary database



All the running services are associated to the primary database. In case of failure of this primary database, all resources are taken back by the secondary database, which makes them available again. Every server instance of the SAP CC system which was connected to the primary database automatically reconnects to the secondary database, in order to ensure the availability of the charging operations.



SAP CC supports this active-passive high availability mode for the following RDBMS:

- SAP HANA, using storage and system replication
- SAP ASE, using the symmetric companion's mode
- Oracle RAC with or without partitioning option
- Microsoft SQL Server, using an external clustering software
- IBM DB2, using the HADR (high availability disaster recovery) feature

High Volume Management

SAP Convergent Charging implements a high volume management feature which concerns:

- The ability to handle intensive incoming traffic of incoming events without impacting the global performance, thanks to routing and caching mechanisms applied on the different instances of the Core Server system
- The possibility to store huge volumes of data, thanks to database optimizations such as data partitioning, connections pooling, and so on
- The ability to perform operations which concern large quantities of elements, such as batch charging operations, recharging operations, and so on

Routing Mechanisms

The architecture of SAP CC relies on a set of server instances (Dispatchers, Updaters, Raters, and so on) which communicate together and share technical and functional information related to the overall business.

To optimize and speed up the dialog between the different initialized server instances, partitioning, round-robin and multicast routing internal mechanisms have been implemented.

Partitioning Routing

The partitioning routing mechanism consists in sending a message to a specific Rater according to a partition identifier. The relationship between Raters and partition identifiers is named "Partition Map", and is handled by Dispatchers using a cached structure. This cached structure contains a list of partitions identifiers:

- Synchronized within all the running Dispatchers
- Used to guide incoming requests to the adequate partition holder



NOTE

The partition identifier is determined by the contacted Dispatcher according to the user and service identifiers provided within the incoming request.

Partitions repartition

SAP Convergent Charging has been designed to handle up to 500 million of subscribers, which represents large amount of associated data. To ensure good performances of the solution, data are partitioned through 480 partitions for rater instances, and 240 partitions for guider instances. These partitions are distributed by the master Dispatcher server instance according to the following algorithm that depends on the number of running Raters and Guiders.

Considering:

- P as the number of partitions to distribute
 - P = 240 in the case of partition distribution among Guider instances
 - P = 480 in the case of partition distribution among Rater instances
- I as the total number of available instances

You can use the following formula: **$P = (P_1 \times I_1) + (P_2 \times I_2)$**

Where each value depends on the distribution of the P partitions:

- **Even distribution** (i.e. P is divisible by I)
 $P_2 = 0 = I_2$

$$P_1 = (P / I_1) \text{ with } I_1 = I$$

■ **Uneven distribution** (i.e. P is not divisible by I)

$$P_1 = \text{Floor} (P / I)$$

$$P_2 = P_1 + 1$$

$$I_2 = P - (P_1 \times I)$$

$$I_1 = I - I_2$$



EXAMPLES

Example 1 (Even distribution): Considering P=240 partitions to distribute among I=15 Guider instances:

■ P_1 is divisible by 15

■ $P_1 = 240 / 15 = 16$

Which leads to the following repartition: (15 Guiders X 16 partitions).

Example 2 (Uneven distribution): Considering P=480 partitions to distribute among I=19 Rater instances:

■ P is not divisible by 19

■ $P_1 = \text{Floor} (480 / 19) = 25$

■ $P_2 = 25 + 1 = 26$

■ $I_2 = 480 - (25 \times 19) = 5$

■ $I_1 = 19 - 5 = 14$

Which leads to the following repartition: (14 Raters X 25 partitions) + (5 Raters X 26 partitions).

For information purpose, the following table contains the partitions repartition for a landscape containing up to 20 Raters or Guiders:

Nb of instances	Repartition of 480 partitions for Raters		Repartition of 240 partitions for Guiders	
	Nb of Raters	Partitions per Rater	Nb of Guiders	Partitions per Guider
1	1	480	1	120
2	2	240	2	120
3	3	160	3	80
4	4	120	4	60
5	5	96	5	48
6	6	80	6	40
7	3	68	5	34
	4	69	2	35
8	8	60	8	30
9	6	53	3	26
	3	54	6	27
10	10	48	10	24
11	4	43	2	21
	7	44	9	22
12	12	40	12	20
13	1	36	7	18
	12	37	6	19

14	10	34	12	17
	4	35	2	18
15	15	32	15	16
16	16	30	16	15
17	13	28	15	14
	4	29	2	15
18	6	26	12	13
	12	27	6	14
19	14	25	7	12
	5	26	12	13
20	20	24	20	12

Partitions mapping reorganization

When a rater or guider is added or removed from the list of running SAP CC system instances, the master dispatcher must reorganize the Partition Map by reallocating the list of partitions among the available instances in order to distribute the overall workload.

The reorganization of the instance map depends on the instance status:

- When an instance stops (or fails), all its handled partitions are reassigned to the other instances of the same type. The PCN (Partition Change Number) of each reassigned partition is increased to inform the instances about the change
- When a new instance starts, the master dispatcher assigns to this new instance a set of partitions, whose PCN is the lowest one to ensure that:
 - These assigned partitions correspond to the coldest ones
 - A restarting instance recovers its previously allocated partitions

The master dispatcher can distribute partitions according to the following execution modes:

- **Normal** mode, which consists in assigning (or reassigning) **a whole set of partitions** to an instance
- **Smart** mode, which consists in assigning (or reassigning) **progressively** the set of partitions to an instance by:
 - First assigning to the instance an initial subset of the computed set of partitions to distribute
 - Assigning the other partitions as time goes by, until the whole set of partitions initially computed has been distributed

For further information, refer to the description of the “Tuning: Partitions Distribution and Warmup” group in the [SAP Convergent Charging System Parameters Reference](#) documentation.

As of SAP Convergent Charging 5.0 SP4SAP Convergent Charging, a new feature named "System Maintenance Mode" has been introduced to optimize these partitions distribution operations performed by the master dispatcher. This feature gives the possibility to switch your Core Server system to a specific running mode within which raters and guiders can be switched to a standby mode, taking into consideration the fact that:

- Every running rater or guider can be switched to a standby mode
- Every starting rater or guider starts in a standby mode
- Every standby rater or guider:
 - Is alive but does not perform any business operation
 - Does not handle any partition but is available for receiving partitions from the master dispatcher
 - Can be manually resumed by the administrator

- Is automatically resumed when the Core Server system exits the maintenance mode
 - Is automatically resumed when another instance stops (and receives the partitions previously handled by this stopped instance)
 - Is not automatically resumed in case of another instance failure (which means that the remaining started instances will handle the partitions previously handled by the crashed instance)
- Every stopping rater or guider preferably transfers its partitions to a standby instance, which is automatically resumed (except when the number of stopped instances exceeds the number of available standby instances, in which case the regular smart warmup mechanism applies)

Therefore, restarting operations of raters and guiders that are performed using standby instances thus:

- Optimize the usage of the available network resources, by decreasing:
- The number of transferred partitions
 - The solicitation of the databases
- Which both decrease the probability of latency peaks
- Decrease the duration of warm-up operations of the cached structures handled by these instances in case of partitions redistribution

For further information, refer to the “Managing the System Maintenance Mode” and “Standing by instances” sections of the Cockpit user interface available in the [SAP Convergent Charging User Interfaces](#) documentation.

Whatever the situation is, every reassigned partition leads to the reset of the cached structures containing the data of this partition. When the cache warmup mechanism is enabled, every reassigned partition must thus be warmed by the targeted instance, which impacts the global performances of SAP Convergent Charging.

To control these warm-up operations, you can use:

- The smart distribution mode, that gives the possibility to control the overall impact of warm-up operations on the overall landscape, that can in addition be optimized using standby instances
- The Network Data Transfer mechanism, that gives the possibility to transfer the cached structures directly from one instance to another using the network, and thus to decrease the solicitation of the Core Database and Session Database

Round-robin routing

The internal round-robin routing mechanism consists in forwarding messages to a given concerned instance using a cyclic mode:

- The first message is sent to the first available instance
- The second message is sent to the next available instance
- And so on

Once all the available server instances have been invoked, the first one is re-invoked, and so on.

When a given instance is not reachable, this routing mode gives the possibility to re-emit a request towards another instance, and thus also contribute to the high availability feature.

Multicast routing

The multicast routing mechanism consists in forwarding messages to multiple instances simultaneously. This mechanism is provided by the master Dispatcher which is connected to all the server instances (as all instances must register with it on startup).

Caching Policies

As described in the [description of the different instances](#), most of the available instances handle data structures stored in memory and managed as cached structures. The most-used cached structures concern:

- Accesses related to the end customers
- Charging sessions (and associated histories)
- Pricing catalog master data (offers, charge plans, refill plans, monitoring plans, allowance plans, translation tables, tier tables, mapping tables and classes, pricing macros, and so on) related to the service provider
- Provisioning master data (provider contracts, subscriptions, subscriber accounts, subscriber mapping tables and allowances and shared allowances) related to the end customers
- Taxation data (business data)

To improve the global performance of SAP CC, these cached structures are stored in memory. Some of these structures can be configured to fit specific needs, while some others are automatically sized and loaded (partially or completely) according to the available resources.



NOTE

Depending on the available resources, auto-sized cache may not have enough memory to handle all the required data. Such a situation does not avoid the startup of the services instances, but decreases the global performance of SAP CC (as information must thus be recovered from databases instead of memory). For further information about cache sizing rules and dependencies between the concerned configuration parameters, refer to the [SAP Convergent Charging Configuration Guide](#) and [SAP Convergent Charging Parameters Reference](#) documentations.

Configurable caches

Some caches used by the different server instances can be configured to fit specific needs and tune the global performance of SAP CC.

Accesses cache

Accesses data are used to determine routing data required to transfer incoming requests to the adequate server instance. To give the possibility to manage an unlimited number of accesses, these data have been partitioned using a hash function applied on the user service identifier property.

Most of the time, all the accesses are not used and keeping all of them in memory is thus superfluous. To reduce memory consumption and improve performances, the list of accesses is henceforth managed using a LRU policy. This LRU mechanism gives the possibility to:

- Place at the beginning of the list the last recently used access
- Remove (from the tail of the list) the oldest used accesses when the cached structure is full and needs free space in order to add a newly used access
- Avoid access creation failure



NOTE

Peaks of latency may occur in the following situations:

- When an incoming request concerns an access which is not in the cache
- When the cache is too small and thus needs to be frequently reorganized
- When a new Guider registers, as this registration implies a cache reorganization

Provisioning cache

The provisioning cache can be configured through different parameters. These parameters are:

- `SUBSCRIPTION_CACHE_INSTANCES`, which defines the maximum number of caches used by each concerned instance. This value is automatically computed on startup to start the appropriate number of instances regarding the managed data
- `SUBSCRIPTION_CACHE_SIZE`, which defines the memory allocated to store the cached objects (provider contracts, subscriptions, counters and subscriber accounts). It is recommended to set a size which is big enough to contain all the managed data. Otherwise, concerned instances have to retrieve data from the database which leads to latencies



NOTES

- To reduce memory consumption and improve performances, the provisioning cache is managed using a LRU policy described above
- The provisioning cache managed by Updaters does not contain counters

Charging sessions cache

This cache structure contains:

- The data for all the charging sessions created during the execution of the [Chargeable Items Charging process](#) performed in session-based mode, for all the subscriber accounts managed by a given Rater
- The history of the charging sessions

The charging sessions cache is also replicated and synchronized in the database by a background mechanism which ensures high availability for session-based charging services:

- This backup copy is used in case of a general crash of a Rater: the other running Raters can retrieve its managed data in order to avoid any service failure
- This backup copy is used in case of a start of a new Rater: this Rater retrieves the data related to the active charging sessions associated to the subscriber accounts it has to manage



NOTE

To reduce memory consumption and improve performances, the charging session cache is managed using a LRU policy described above

The use of this structure can be configured through different parameters. These parameters are:

- `SESSION_MEMORY_INSTANCES`, which defines the maximum number of caches which can be used by each instance. This value is automatically computed on startup to start the appropriate number of instances regarding the managed data
- `SESSION_MEMORY_SIZE`, which defines the memory allocated to each instance to store the cached objects. It is recommended to set a size which is big enough to contain all the charging sessions to manage. Otherwise, concerned instances have to retrieve data from the database which leads to latencies

Shared Allowances cache

This cache structure contains:

- The shared allowances (that represent allowances shared between subscriber accounts)
- The distributed counters (that correspond to persistent counters of shared allowances)

 **NOTE**

To reduce memory consumption and improve performances, the shared allowances cache is managed using a LRU policy described above

The use of this structure can be configured through the following system parameters. For further information, refer to the description of the “Process: Allowance Management” group available in the [SAP Convergent Charging System Parameters Reference](#) documentation:

- SHARED_ALLOWANCE_CACHE_REFRESH_SCHEDULER_ENABLED
- SHARED_ALLOWANCE_CACHE_REFRESH_SCHEDULER_RECURRENCE
- SHARED_ALLOWANCE_CACHE_REFRESH_SCHEDULER_LAST_TRIGGER
- SHARED_ALLOWANCE_CACHE_INSTANCES
- SHARED_ALLOWANCE_CACHE_SIZE
- SHARED_ALLOWANCE_OBJECT_AVERAGE_SIZE
- SHARED_ALLOWANCE_CACHE_FREE_MEMORY
- SHARED_ALLOWANCE_CACHE_STATUS
- DISTRIBUTED_COUNTER_RESERVATION_POOL_SIZE
- DISTRIBUTED_COUNTER_RESERVATION_MAX_BATCH_SIZE
- DISTRIBUTED_COUNTER_RESERVATION_QUEUE_POLICY_HASH_KEY_SIZE
- DISTRIBUTED_COUNTER_RESERVATION_QUEUE_POLICY_LEVELS
- DISTRIBUTED_COUNTER_CLEANUP_POOL_SIZE
- DISTRIBUTED_COUNTER_CLEANUP_MAX_BATCH_SIZE
- DISTRIBUTED_COUNTER_CLEANUP_QUEUE_POLICY_HASH_KEY_SIZE
- DISTRIBUTED_COUNTER_CLEANUP_QUEUE_POLICY_LEVELS
- DISTRIBUTED_COUNTER_CHECK_POOL_SIZE
- DISTRIBUTED_COUNTER_EMPTY_CHECK_INTERVAL
- DISTRIBUTED_COUNTER_EMPTY_CHECK_MAX_THROUGHPUT

Auto-sized caches

Some cached structures of SAP CC are automatically sized according to the available resources.

Catalog cache

Also called offer-level cache, the catalog cache contains objects such as:

- Master data
 - Offers
 - Charge plans
 - Charges
 - Refill plans
 - Refill logic objects
 - Allowance plans
 - Allowance logic objects
 - Monitoring plans
 - Pricing macros
 - Translation tables
 - Tier tables

- Mapping tables
- Range tables
- And so on
- Business data
 - Public holidays

As these objects correspond to business master data, the corresponded cached structures are entirely loaded on startup, in the first position in the cache loading list.

 **NOTE**

The catalog cache works with a flip-flap mechanism: when an Updater modifies an object, a multicast notification is sent to the running Raters. Concerned Raters then update a copy of the managed catalog cache, until a flip-flap is performed (which consists in overwriting the managed cache with the updated copy). This mechanism can be manually or automatically performed.

Taxation data cache

As some instances need to quickly access to taxation data (such as VAT taxes and VAT rules), a cached structure is implemented to store this business data in memory. This structure is entirely loaded on startup, and is thus not configurable.

Cache warm-up

To avoid latency and increase the reactivity of the Core Server, a warm-up mechanism has been implemented for the cached structures handled by raters and guiders. This mechanism consists in:

- Retrieving all the data related to vendors and clients, either from database or directly from another instance (depending whether the Network Data Transfer mechanism is enabled or not)
- Loading this data into the related cached structures handled by the concerned instances

The cache warm-up mechanism uses an intelligent loading strategy which ensures that the cached structures only contain the last versions of the master data (whatever the operation which could impact a master data is). A dedicated parameter named "CACHE_WARMUP_THREAD_COUNT" gives the possibility to:

- Deactivate the mechanism, for business or technical reasons such as backward compatibility, limited available memory, moderated business, and so on
- Activate the mechanism and associate a number of dedicated threads in order to tune the performance of the mechanism's execution (warm-up speed, cached structures availability, database stress, and so on)

Admin+ contains a command named "start_cache_warmup" that gives the possibility to start and/or restart the cache warm-up mechanism on a given running instance (for example after operations such as data migration, mass provisioning, and so on, which lead to lots of invalidations of the cached data).

 **NOTE**

For further information about the "CACHE_WARMUP_THREAD_COUNT" parameter and the `start_cache_warmup` command, refer to the [SAP Convergent Charging Configuration Guide](#) and [SAP Convergent Charging Parameters Reference](#) documentations.

Network Data Transfer

The Network Data Transfer mechanism gives the possibility to transfer high volume of data between instances within your SAP Convergent Charging landscape.

This mechanism positively impacts the performance of your SAP CC landscape by:

- Transferring cached data over the network directly between instances, which decreases the overall solicitation of the Core and Session databases during warmup operations
- Decreasing the duration of warm-up operations of the cached structures handled by raters and guiders, thanks to a better usage of network resources in case of partitions redistribution

The Network Data Transfer mechanism relies on 2 different types of messages (requests and responses) used by both raters and guiders:

- **Class1 messages**, that correspond to data transfer requests and responses performed during priority operations such as real time operations, administrations queries during data transfer operations, and so on
- **Class2 messages**, that correspond to data transfer requests and responses performed during non-priority operations such as data transfer operations in a context of cache warmup operations

For optimization purpose, it is also possible to control the behavior of these messages during data transfer operations by configuring:

- The resources allocated to raters and guiders when processing Class1 or Class2 messages (number of threads, size of queues)
- The bandwidth allocated to raters and guiders when dealing with Class2 messages, to avoid interfering with real time services like stateful charging
- The size of the Class2 messages, to avoid overloading the network

For further information about the configuration of the Network Data Transfer mechanism, refer to the "Enabling the network data transfer mechanism" section of the [SAP Convergent Charging Tuning Guide](#) documentation.

Synthesis

The following table shows the managed structures according to the different instances:

Data structure	Dispatcher	Rater	Guider	Updater	Taxer	Bulkloader
Technical structures						
Custom audit filters	■	■	■	■	■	■
SAP CC users	■	■		■		
Partitions	■					
Accesses			■			
Charging sessions		■				
Taxation data		■		■	■	
Batch rating groups				■		
Exportable item mapping		■		■		■
Provisioning data structure						
Provider contracts		■	■	■		
Allowances		■		■		
Subscriptions		■	■	■		
Subscriber accounts		■		■		
Counters		■		■		

Subscriber mapping tables and subscriber range tables		■	■	■		
Catalog data structure						
Offers		■		■		■
Charges		■		■		
Refill logic objects						
Allowance logic objects						
Charge plans		■		■		
Refill plans						
Allowance plans						
Monitoring plans						
Chargeable item classes and packages		■		■		■
Refill item classes		■		■		
Allowance event classes		■		■		
Allowance interface		■		■		
Translation tables, mapping tables and classes		■		■		
Tier tables, range tables and classes		■		■		
Pricing macros		■		■		
Charged item classes		■		■		■
Refill record classes						
Aggregation policies						
Public holidays		■		■		

Batch rating group management

Some operations such as batch charging or recharging operations can concern large quantities of elements. To ease the execution and increase the performance of such operations, it is possible to group the concerned data according to batch rating groups.

These batch rating groups:

- Must first be defined into the system, using the `groupadd`, `groupdel`, `grouplist` and `groupmod` commands of Admin+
- Can be assigned to provider contracts or offline subscriptions, at creation time
- Are linked to the CDRs during the execution of the [Chargeable Items Acquisition Process](#) performed by the BART Server, to group the CDR for business or technical purposes (such as database partitioning for better performances during batch charging operations)

The assignment of batch rating group is different whether it concerns subscriptions or provider contracts:

- To assign a batch rating group to an offline subscription, it is just necessary to select one of the existing batch rating group, at creation or modification time
- The assignment of batch rating groups for provider contracts can only be performed at creation time, when provider contracts are replicated from SAP CRM (which does not manage technical information such as batch rating groups). The "BATCH_RATING_GROUP_ASSIGNMENT_POLICY_CLASS" configuration parameter gives the possibility to define the rules which must be applied in order to assign batch rating groups to the replicated

provider contracts. A default implementation of the assignment policy is provided, and balances the provider contracts on the defined batch rating groups. It is possible to adapt this behavior to specific needs and thus implement specific rules, but requires specific developments as described in the example afterwards



NOTES

- It is possible to update the batch rating group of a given offline subscription, **but it is highly not recommended to do such a modification**, as it can change the expected behavior of further batch operations
- For further information about the commands of Admin+ dedicated to batch rating groups management, refer to the [SAP Convergent Charging Configuration Guide](#) documentation
- For further information about the “BATCH_RATING_GROUP_ASSIGNMENT_POLICY_CLASS” parameter, refer to the “Updater” section of the [SAP Convergent Charging Parameters Reference](#) documentation



EXAMPLE

Considering two existing batch rating groups named “Group 1” and “Group 2”, respectively defined with the “1” and “2” identifiers, a customer wants to allocate the provider contracts whose identifiers end with “0”, “2”, “4”, “6”, “8” to the “Group 1” batch rating group, and the other provider contracts to the “Group 2” batch rating group. To implement such an assignment policy rule, it is necessary to:

- Create a specific java class which implements the `IContractBatchRatingGroupAssignmentPolicy` interface and overrides the `getBatchRatingGroupAssignment(...)` method, for example:

```
package sap.cc.integration.batchratinggroup;

import com.highdeal.batchgroup.BatchRatingGroupIdentifier;
import com.highdeal.batchgroup.BatchRatingGroupIdentifierException;
import com.highdeal.batchgroup.IBatchRatingGroupsAccessor;
import com.highdeal.batchgroup.contract.IContractBatchRatingGroupAssignmentPolicy;
import com.highdeal.contract.hci.ChargingContractRevisionModel;

public class BatchRatingGroupImpl implements IContractBatchRatingGroupAssignmentPolicy {

    public BatchRatingGroupIdentifier getBatchRatingGroupAssignment(
        ChargingContractRevisionModel arg0, IBatchRatingGroupsAccessor arg1)
        throws BatchRatingGroupIdentifierException {

        if ( arg0.getId().endsWith("0") || arg0.getId().endsWith("2") ||
            arg0.getId().endsWith("4") || arg0.getId().endsWith("6") ||
            arg0.getId().endsWith("8") )
            return arg1.getBatchRatinggroupById((short) 1);
        else
            return arg1.getBatchRatinggroupById((short) 2);
    }
}
```

- Create a JAR file containing this single class, and insert this JAR into the folder which contains the JAR files which must be copied to the instance local folder at startup (such as the `/usr/sap/XXX/SYS/exe/uc/XXXX/CC_CORE_SERVER/jars` folder)
- Modify the value of the “BATCH_RATING_GROUP_ASSIGNMENT_POLICY_CLASS” parameter in order to use your own implementation (using the `<package_name>.<class_name>` format)
- Use the Config tool to update the system configuration and thus take into account this new assignment policy
- Add the created JAR file to the classpath of the Updater, described in its associated `jstart.config` file
- Restart the system

System and Data Auditing

SAP Convergent Charging provides the System and Data Auditing feature to fulfill your business and technical requirements relating to the system audit and control aspects and to implement your security policy in the system landscape.

SAP CC delivers the following fundamental features:

- Master data and user operation auditing
- Product usage measurement

To fulfill specific auditing and control requirements, you can implement the following technical features and functions in the SAP CC Core Server system:

Technical Feature	Core Server	Core Tool or Cockpit
Master Data Auditing		
Recording of object change logs	■	
Inspecting of object change logs (data object changes)		■
Recording of object snapshots	■	
Inspecting of object snapshots (data detailed changes)		■
User Operation Auditing		
Recording of requested user operations	■	
Inspecting of audited user operations		■
Product Usage Measurement		
Product Usage Metrics	■	■
GLAS Files		■
Customer Usage in SAP Solution Manager	■	

- Automated functions
- Manual tasks with tools

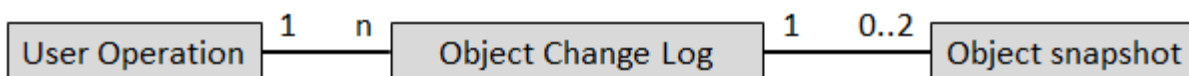
Master Data and User Operation Auditing

To manage the auditing and control levels in your system landscape, SAP Convergent Charging provides the Master Data and User Operation Auditing feature to track the modifications of master data in the back-end database from an end-to-end prospective. The SAP CC Core Server system automatically audits:

- Modifications of master data
- SAP CC users and user operations that are responsible for these modifications

SAP CC stores the following audit information into the Core database:

- **Object change logs**, which correspond to recordings of modifications performed on data objects
- **Object snapshots**, which correspond to copies of a data object before and after the modifications
- **Audited user operations**, which correspond to recordings of successful executions of requested user operations or recordings of failed operations



This fine-grained auditing mode provides you and your SAP Support Team with an efficient supportability tool to audit and analyze data modifications and user operations executed within your landscape.

SAP CC gives the possibility to audit the following data objects:

- Pricing catalog data (pricing macros, charges, charge plans, range tables, and so on)
- Customer master data (subscriber accounts, provider contracts, subscriptions, accesses, subscriber range tables, and so on)
- Business and custom data (currencies, public holidays, chargeable item mapping, and so on) configured in the complete system and shared by all the pricing catalogs and customer master data
- Technical data (SAP users, custom audit filters, and so on)

The auditable user operations are a customized selection of user operations requested via the Web Services and HCI technical interfaces, that modify the system state or the persistent configuration data.

In the Core Tool user interface, only SAP CC users granted the "Administrator" or "Remote Support" role can inspect the audited user operations, object change logs, and object snapshots. For more information about these roles and associated authorizations, refer to the [SAP Convergent Charging Security Guide documentation](#).



RECOMMENDATION

By default, this end-to-end auditing is enabled. SAP recommends that you plan to implement and customize the following technical features:

- Recording of object change logs
- Recording of object snapshots
- Recording of user operations

Auditing functions can lead to the creation of large quantities of data stored in the back-end database that can significantly impact the performance and storage capacity of the database. SAP recommends that you:

- Optimize your implementation by defining the auditable objects and operations, the database sizing and to reserve space accordingly
- Plan to archive and purge audit data recordings from the database regularly

Recording of Object Change Logs

SAP Convergent Charging gives the possibility to log the modifications on persistent configuration data (master data, business/custom data). It supports data creation, modification, and deletion requested via the Web Services and HCI technical interfaces and via the SAP CC Core Tool, Admin+ and Setup Tool user interfaces.

The SAP CC Core Server system automatically audits the modified data objects by recording an object change log into the Core database within a reserved space.

The SAP CC Core Tool user interface gives the possibility to view and inspect the object change logs in audit recordings. Only an SAP CC user granted the "Administration" or "Remote Support" role can access such information



NOTE

For security reasons, this technical feature is always enabled, which means that you cannot deactivate the feature and its associated functions



CAUTION

The auditing functions can lead to the creation of large quantities of audit data recordings stored in the OBJECT_CHANGE_LOG table of the Core database.

To run your system optimally, SAP recommends that you manage the data volume of audited information by:

- Sizing or resizing the database by considering the OBJECT_CHANGE_DATA reserved space
- Regularly planning archiving and purging operations for out-of-date object change logs from audited data recordings, using the `purge_object_change_logs` command of the SAP CC Admin+ user interface



For further information, refer to the [SAP Convergent Charging Configuration Guide](#), [SAP Convergent Charging Operations Guide](#) and [SAP Convergent Charging User Interfaces](#) documentations.

Audited information

The following table contains the list of possible audited information:

Audited Information	Description
Identification	The necessary identifiers of the concerned audited data object (possibly a multi-part identifier)
Date and time	The date when the data object has been modified
Object type	The type of concerned data objects (user, currency, charge plan, range table, chargeable item class, and so on)
Treatment	The type of performed action (creation, modification, or deletion)
Supportability tool	The technical and unique reference of the object change log stored in the Core database, that may help your support specialist. This object unique ID (OID) is automatically determined by SAP CC.

Associated to the recording of user operations and object snapshots, the object change logs provide additional audit information such as:

Audited Information	Description	Prerequisites
User operation	<p>The reference of the audited user operation.</p> <p> NOTE This important information can identify the SAP CC user who requested the SAP system to process the operation</p>	The recording of user operation is enabled in the SAP CC Core Server system
Data object values (in snapshots)	<p>The references of the copies of the data object before and after the modifications of the object.</p> <p> NOTE This important information gives the possibility to analyze the details of the object modification</p>	The recording of object snapshots is enabled in the SAP CC system

Auditable data objects

The following table contains the list of data which support the auditing function:

Business and custom data	Pricing catalog data		Customer master data	Technical data
Allowance Interface Counter Names Dictionary	Pricing Macro	Chargeable Item Package	Subscriber Account	User
Currencies	Translation Table	Charged Item Class	Subscription	Audit (Custom filters)
ISO 4217 Currencies	Tier Table	Aggregation Policy	Provider Contract	
Public Holidays	Mapping Table Class	Refill Item Class	Access	
Exportable Item Mapping	Mapping Table	Refill Record Class	Subscriber Mapping Table	
	Range Table Class	Refill Logic	Subscriber Range Table	
	Range Table	Refill Plan		
	Offer	Allowance Plan		
	Charge	Allowance Logic		
	Charge Plan	Allowance Event Class		
	Chargeable Item Class	Monitoring Plan		

Recording of Object Snapshots

SAP Convergent Charging gives the possibility to enrich audits with detailed information about the modifications performed on the audited data objects.

The SAP CC Core Server system audits a customized selection of modified data objects by recording object copies respectively before and after each modification. The SAP CC system stores these object snapshots within the Core database in a reserved space. You can fine configure the categories of data objects that can be audited to object snapshots. SAP CC defines some useful audit domains that facilitate the system configuration.

The SAP CC Core Tool user interface gives the possibility to view and inspect the object snapshots in audit recordings. Only an SAP CC user granted the "Administration" or "Remote Support" role can view this information.



CAUTION

The auditing functions can lead to the creation of large quantities of audit data recordings stored in the OBJECT_SNAPSHOT table of the Core database, which may significantly impact the performance of the complete SAP CC landscape.

To run your system optimally, SAP recommends that you manage the data volume of audited information by:

- Defining the audit domains that fit your technical and business requirements
- Sizing or resizing the database by considering the OBJECT_CHANGE_DATA reserved space
- Regularly planning archiving and purging operations for out-of-date snapshots and object change logs from audited data recordings, using the `purge_object_change_logs` command of the SAP CC Admin+ user interface

For further information, refer to the [SAP Convergent Charging Configuration Guide](#), [SAP Convergent Charging Operations Guide](#) and [SAP Convergent Charging User Interfaces](#) documentations.

Audit domains and associated data types

When you select an audit domain, SAP CC only records the corresponding object snapshots everytime an object is created, modified or deleted by an SAP CC user.

The recording of object snapshots can be customized by defining the audit domains that satisfy your business and technical requirements (security, audit level, or database storage and performance).

The following table contains the list of available audit domains:

Audit Domain	Audited Data Type
CATALOG	Data objects in pricing catalogs (in Master Data) extended with business and custom data such as the currencies or the counter name dictionary
CATALOG_TABLE_DATA	Business data tables in pricing catalogs (mapping tables, range tables, tier tables, and translation tables)
CUSTOMER	Customer master data (in Master Data) such as provider contracts, subscriber accounts, or accesses
CUSTOMER_TABLE_DATA	Business data tables in customer master data: subscriber mapping tables and subscriber range tables
TECHNICAL	Technical configuration data such as the SAP CC users or the custom audit filters

The following table details the list of data that can be audited, including the associated audit domains:

Audited Data Types	Audit Domains				
	CATALOG	CATALOG_TABLE_DATA	CUSTOMER	CUSTOMER_TABLE_DATA	TECHNICAL
Master Data					
Pricing Catalog Data					
Aggregation Policy	■				
Allowance Event Class	■				
Allowance Logic	■				
Allowance Plan	■				
Charge	■				
Charge Plan	■				
Chargeable Item Class	■				
Chargeable Item Package	■				
Charged Item Class	■				
Mapping Table		■			
Mapping Table Class	■				
Monitoring Plan	■				
Offer	■				
Pricing Macro	■				
Range Table		■			
Range Table Class	■				
Refill Item Class	■				
Refill Logic	■				
Refill Plan	■				
Refill Record Class	■				
Tier Table		■			
Translation Table		■			
Customer Master Data					
Access			■		
Provider Contract			■		
Subscriber Account			■		
Subscriber Mapping Table				■	
Subscriber Range Table				■	
Subscription			■		
Business Data and Custom Data					

Allowance Interface	<input checked="" type="checkbox"/>				
Counter Names Dictionary	<input checked="" type="checkbox"/>				
Currencies	<input checked="" type="checkbox"/>				
Exportable Item Mapping	<input checked="" type="checkbox"/>				
ISO 4217 Currencies	<input checked="" type="checkbox"/>				
Public Holidays	<input checked="" type="checkbox"/>				
Technical Data					
User					<input type="checkbox"/>
Audit (Custom Filter)					<input type="checkbox"/>

- Master data
- Business and custom data
- Technical data

Recording of User Operations

SAP Convergent Charging gives the possibility to track the user operations requested by the SAP CC users (individual users and service users), whatever the used user interface or technical interface (Web Services or HCI) is.

The SAP CC Core Server system audits all the requested operations by recording their execution and result (success, failure) automatically within the Core database system.

The SAP CC Core Tool user interface gives the possibility to view and inspect the audited operations and to continue your investigations. You can also view the related object change log when the audited operation modified the data object (creation, modification, deletion). Only an SAP CC user granted the "Administration" or "Remote Support" role can view this information.

You can audit:

- The successful operations that modified the persistent configuration data in the Core database
- The successful operations that modified state of the the SAP CC Core Server system
- All the failed operations

For security reason, you can audit the successful search operations that retrieve lists of audited user operations.

You cannot audit charging and refilling operations even if they modify the customer master data. Refer to the audit information specified in the Web Services Documentation. In case of bulk operations, refer to the "Functions and Mechanism Details" section afterwards.



IMPORTANT NOTE

The failure during the auditing of a successful user operation leads to the cancellation of the original user operation. The SAP CC system traces this issue.



CAUTION

The auditing functions can lead to the creation of large quantities of audit data recordings stored in the USER_OPERATION table of the Core database, which may significantly impact the performance of the complete SAP CC landscape.

To run your system optimally, SAP recommends that you manage the data volume of audited information by:

- Defining the operation audit domains that fit your technical and business requirements
- Sizing or resizing the database

- Regularly planning archiving and purging operations for out-of-date snapshots and object change logs from audited data recordings, using the `purge_user_operations` command of the SAP CC Admin+ user interface

For further information, refer to the [SAP Convergent Charging Configuration Guide](#), [SAP Convergent Charging Operations Guide](#) and [SAP Convergent Charging User Interfaces](#) documentations.

Audited information

This technical feature gives the possibility to create some audits to get information such as:

- The comprehensive name of the user operation
- The SAP CC individual or technical user who requested a given operation
- The date when the operation has been executed
- The operand of the processed user operation, that can be an identifier, a reference or an annotation characterizing the user operation or the data object impacted by the operation. For example, it can be the name or identifier of a data object or the technical name of a system parameter

Operation audit domains

The following table contains the list of possible operation audit domains that you can use to fine configure the audit of successful user operations. The selected domains define a set of auditable user operations for which SAP CC creates audit data recordings within the Core database.

Operation Audit Domain	Typical Audited Operations
CATALOG	User operations that modify data objects in: <ul style="list-style-type: none"> ■ Pricing catalogs (associated to the customer services to monetize) ■ Business and custom data such as currencies, public holidays, billable item mapping objects, or counter name dictionary
CUSTOMER	User operations that modify data objects in customer master data such as subscriber accounts, provider contracts, or subscriber mapping/range tables
TECHNICAL	User operations that modify technical data such as SAP users or custom audit filters
ADMINISTRATION	Technical or business operations for administrating or configuring the SAP CC system such as system parameters modifications or activation process triggering

Functions and Mechanism Details

Auditing of Bulk Operations

SAP CC gives the possibility to audit bulk operations as a single operation, using a unique audit recording in the audited user operations. All the configuration data modified by this bulk operation is audited and refers to the same audited operation. For example, you can audit bulk operations such as rerating in bulk or subscription modification in bulk.

Note that the Web Services technical interface provides two particular modes for bulk processing: the “bundle” and “mass” modes. Both modes of user operations group multiple single operations for processing, and the associated auditing mechanism is different. Refer afterwards for details about such processing.

Auditing of Bundle and Mass Operations

When a bundle or a mass operation is audited, each single operation is audited and the corresponding audit data recordings is stored in the Core database. There is no recording that relates to the bundle or mass operation itself.

 **NOTE**

When a bundle operation is audited, the failure of the auditing of a successful single operation leads to the cancellation of this single operation but also to the cancellation of the bundle operation including all its single user operations.

Feature Implementation

The Master Data and User Operation Auditing feature can lead to the creation of large quantities of data stored in the back-end database that can significantly impact the performance and storage capacity of the whole SAP CC Core Server system.

As an IT consultant, you translate business needs into requirements and determine if these technical and business requirements involve this feature. You consider required security, audit and control level, database storage and performance. You may re-consider the sizing of the Core database.

For further information about the system configuration and important recommendations, refer to the [SAP Convergent Charging Configuration Guide](#) and the [System Parameter Reference](#) documentation, particularly the following system parameters:

Technical Feature	Function Enablement	Audit Domains
Master Data Auditing		
Recording of Object Change Logs	Always enabled	
Recording of Object Snapshots	AUDIT_OBJECT_ACTIVATION	AUDIT_OBJECT_DOMAIN
User Operation Auditing		
Recording of User Operations	AUDIT_ACTIVATION	AUDIT_DOMAIN

 **NOTE**

SAP recommends that you perform purge operations manually and regularly in order to optimally maintain and run your SAP CC landscape. For further information about these technical operations, refer to the [SAP Convergent Charging Operations Guide](#) and [SAP Convergent Charging User Interfaces](#) documentations.

Product Usage Measurement

The Product Usage Measurement feature gives the possibility to audit the product usage of SAP Convergent Charging implemented in your system landscape. The SAP product usage includes the measurements of all your productive Core Server systems.

SAP Convergent Charging is part of different SAP System Measurement programs. You use these programs to verify your system usage and to plan some license extension.

The following table contains the principle measurements and statistics:

Area	Product Usage Measure
Business Operations	

Accounting Operations (Convergent Charging)	The volume of transactional data charged by SAP CC (output data items): <ul style="list-style-type: none"> ■ Total number of prepaid account operations generated by the system ■ Total number of postpaid account operations generated by the system
Business Configuration	
Configured Master Data (in High Volume)	The volume of configuration data managed by the SAP CC system: <ul style="list-style-type: none"> ■ Total number of subscriber accounts available in the system

The SAP CC Core Server system measures automatically this product usage information when it charges the end customers for their service usages and when it configures the subscriber accounts in customer master data.

SAP Convergent Charging delivers 3 technical features to manage the audit and control of your business solution:

- Usage Metric Recording and Reporting
- GLAS File Reporting
- Customer Usage Measurement in SAP Solution Manager

Product Usage Metrics

SAP Convergent Charging provides a technical feature that gives the possibility to audit the product usage of the SAP CC software product and systems implemented in your system landscape, which represents the base of all the other product measurement features.

The Core Server system daily records the following usage metrics:

- Daily number of subscriber accounts in the system (creation, deletion)
- Daily number of charged items or charged transactions (accounting operations) processed by the system
- Daily number of prepaid transactions (accounting operations) processed by the system in a prepaid payment environment
- Daily number of postpaid transactions (accounting operations) processed by the system in a postpaid payment environment

This technical feature includes the following functions:

- In the Core Server system, the updater instance collects and consolidates the usage metrics daily
- SAP CC stores an audit data recording into the Core database
- The Cockpit and the Core Tool user interfaces give the possibility to view and report this product usage information

Implementation: Configuration

The implementation represents a basic system configuration. For further information about the internal scheduler function and the METRIC_RECORDING_SCHEDULER_ENABLED system parameter, refer to the [SAP CC Configuration Guide documentation](#).

More information

For further information about the procedure to view metrics, refer to the Cockpit or to the Core Tool online help documentation available in the [SAP Convergent Charging User Interfaces](#) documentation. Refer to the [SAP Convergent Charging Operations Guide](#) documentation for user assistance about the management and auditing of an SAP CC system.

GLAS Files

SAP Convergent Charging supports the SAP GLAS technology to provide the SAP Global License Auditing Services (GLAS) team with license auditing reports as XML files. When required, you can report product usage information associated to an SAP CC system manually.

The Cockpit and the Core Tool user interfaces give the possibility for IT administrators to create a GLAS file that is taken into account by the SAP Global License Auditing Services team.

For a year, this file contains product usage statistics and event dates such as:

- Peak number of prepaid transactions per day
- Peak number of postpaid transactions per day
- Peak number of transactions per day
- Peak number of subscriber accounts available in master data

More information

For further information about the procedure to generate a license auditing file, refer to the Cockpit or the Core Tool online help documentation available in the [SAP Convergent Charging User Interfaces](#) documentation. Refer to the [SAP Convergent Charging Operations Guide](#) documentation for user assistance about the management and auditing of an SAP CC system.

Customer Usage Measurement

SAP Convergent Charging provides SAP Solution Manager with information about product usage of the system landscape. In the SAP CC Core Server system, the updater instance communicates the usage measurement of the installed SAP CC software product to the requester SAP Diagnostics Agent (SAP SMD Agent). The updater instance replies product usage information and statistics based on the usage metrics recorded in the Core database:

SAP Customer Usage Measure	Technical Reference
Total number of prepaid account operations processed by the system	ISCC_PREPAID_CHARGED_ITEM
Total number of postpaid account operations processed by the system	ISCC_POSTPAID_CHARGED_ITEM
Total number of subscriber accounts in the system	ISCC_SUBSCRIBER_ACCOUNT



NOTES

- The Customer Usage Measurement feature is not available if HTTP communications with the updater instances are secured
- This technical feature requires that the recording of product usage metrics is enabled in SAP CC
- Refer to the product documentation and user assistance of SAP Solution Manager for more information about the available and necessary technical operations

Implementation: Configuration

The implementation represents a basic system configuration in both SAP CC and SAP Solution Manager. For further information about the related configuration, refer to the [SAP CC Configuration Guide](#) documentation.

Data Purging

As described above, SAP Convergent Charging implements a high volume management feature which gives the possibility to store large quantities of data relating to a running SAP CC system.

To ensure the global performance of the system and avoid latencies due to these large quantities of data, some technical mechanisms have been implemented to give the possibility to remove obsolete and/or useless data. These mechanisms concern the following data stored in the back-end databases:

- Audit data recordings, which relate to the auditing features
- Idempotency data, which concern the execution of Web Services operations
- Rerating sessions data, which concern BART rerating sessions which did not closed properly
- Allowances data, which concern the expired allowances which have exceeded a given retention period
- Change lists data, which concern the change lists successfully transported to a remote SAP CC system

Allowances Purge

To reduce the memory footprint of the Allowances Management feature, SAP CC gives the possibility to purge data related to expired allowances, which means allowances whose validity end date (shifted by the retention period) is anterior or equals to the date of the purge operation. Please note that for business purposes, it is possible to configure a retention period during which expired allowances are excluded from the scope of purge operations.

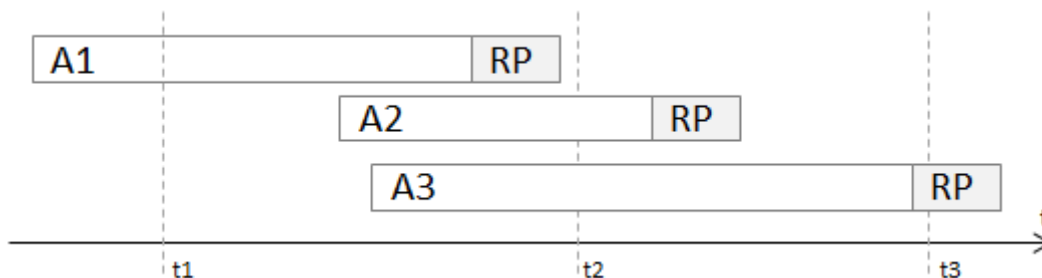
This purge mechanism can be executed:

- Automatically using a dedicated scheduler
- Manually using a dedicated command of the Admin+ user interface

For further information about the execution of this purge mechanism, refer to the SAP Convergent Charging Configuration Guide and SAP Convergent Charging User Interfaces documentations.

EXAMPLE

Considering the following A1, A2 and A3 allowances, and a retention period RP:



At t1 time, the purge operation cannot take any allowance into account. At t2 time, the A1 allowance is expired and can thus be purged. At t3 time, the A1 and A2 allowances are expired and can thus be purged, but not the A3 allowance because of the retention period which is not ended.

RECOMMENDATIONS

Purge operations are executed by the Rater instances. It is thus highly recommended to execute these operations when the system is less solicited, in order to limit the impact on the performance of the charging operations. The following 2 parameters are dedicated to purge operations and give the possibility to control the Raters' solicitation:

- ALLOWANCES_PURGE_RETENTION_PERIOD, which corresponds to the number of days until which allowances whose validity end date is exceeded are kept into the back-end database and cannot be deleted by allowances

purge operations. The longer this retention period is, the more allowances are concerned by purge operations, and thus the more the Raters are solicited

- ALLOWANCES_PURGE_THREAD_COUNT, which gives the possibility to specify the number of threads that the Raters can use when performing allowances purge operations

For further information about these parameters, refer to the [SAP Convergent Charging Parameters Reference](#) documentation.



CAUTION

It is not possible to undo a purge operation related to expired allowances.

Change Lists Purge

To reduce the memory footprint of the [Catalog Transport](#) feature, SAP CC gives the possibility to purge data related to transported change lists, which means change lists whose associated transport requests have an “executed” status. This purge mechanism can be manually executed using the dedicated “purge_change_lists” command of the Admin+ user interface.

For further information about the execution of this purge mechanism, refer to the [SAP Convergent Charging User Interfaces](#) documentation.

Enhanced Logging and Tracing Framework

Logging and tracing are integral features of SAP Convergent Charging for extended supportability. Logging and tracing is essential to analyze problems inside or between systems and applications. Every SAP CC system or graphical user interface records event information or error in:

- Log messages, used to permanently record the events and status of a system in order to be able to determine corrective or preventive actions by IT administrators and operation teams
- Trace messages, used to analyze a given system in details from a support specialist standpoint (as an exceptional procedure for troubleshooting or debugging purposes)

To facilitate information requirements for different levels of troubleshooting, log messages are recorded by categories and trace messages by technical component locations and tracing domains.

You can define the levels of these generated logs and traces using thresholds based on the severity levels of these messages. It is also possible to configure several destinations of these records in order to view and analyze them or to send them to a particular monitoring system such as SAP Solution Manager. The following table summarizes these default settings:

Data	Default Destination	Console	Severity Threshold	File Rotation
Logs	Fileset #1	Log Disabled	INFO	Enabled
Traces	Fileset #2	Trace Disabled	ERROR	Enabled



NOTES

- By default, the logging and tracing functions are disabled in the graphical user interfaces of SAP CC

- The Cockpit application deployed on a Tomcat Server only generates trace messages

For further information, refer to the [SAP Convergent Charging Parameters Reference](#) documentation and to the [SAP Convergent Charging Configuration Guide](#) documentation.

Logging

Logging is a process which consists in creating and storing permanent records related to events, which can be reviewed, printed, and analyzed by administrators. Each log is unique and identified by a message identifier dedicated to the SAP CC software (see “IS-CC” component). A complete reference of log messages details the logs, explains the possible reasons, and provides possible troubleshooting actions for an IT administrator and the operation team. The following table summarizes these messages:

Typical Log	Description	Action	Log Severity
Error messages	An error log message is generated when the processing of a function cannot proceed and a corrective action thus needs to be taken by an administrator. The log describes the problem, the possible impact and suggests resolutions actions.	Corrective	<ul style="list-style-type: none"> ■ Fatal ■ Error
Event notifications	If there are occasions in the software or hardware components, which carry probably important information for the addressees (administrators, support specialists), but which do not need immediate (or possibly none at all) corrective action, this is called “event notification”. The log describes the issue, the possible impact and suggests resolutions actions.	Preventive	<ul style="list-style-type: none"> ■ Warning ■ Info
Status recording	An informational log is generated to describe the status of the system, its subcomponents and the processing functions: <ul style="list-style-type: none"> ■ Startup and shutdown of software component or system ■ Information about the progress of an operation request, a function or a process 	N/A	<ul style="list-style-type: none"> ■ Info
Context information	A second log is generated in conjunction with another log message to provide additional information.	N/A	<ul style="list-style-type: none"> ■ Info

Log records provide the following common information:

- A short descriptive message
- A timestamp of the event
- The source of the record
- The log controller
- A severity, specifying the importance of the record
- The host, system and instance names
- The server process

Log Severities, Thresholds, and Filtering

An important part of any log message is its own severity. It denotes the level of importance or relevance of a certain message. Logs can be restricted to certain severity levels, that is, only log event of a defined severity is generated and collected. The increasing order of the severity levels are:

Log Severity	Description	Follow-Up Activity and Troubleshooting	Examples
Info	Informational text, mostly to record an event or a status change of the SAP CC Core Server system or of a processing function. It must help to understand the normal operation or can provide further information for later reviews, audits or investigations.	No follow-up activity is needed.	<ul style="list-style-type: none"> ■ Event notifications ■ Status recording ■ Context information
Warning	The SAP CC Core Server system or a subcomponent can recover from anomaly, and fulfill the desired task, but needs later follow-up activity by a system administrator or an SAP CC administrator to avoid erroneous terminations in the future.	Follow-up activity to take into account the anomaly and to determine the appropriate timeframe to perform preventive actions.	<ul style="list-style-type: none"> ■ Event notifications
Error	The system or a subcomponent can recover from error, but cannot complete the desired task due to the error. The system or the software application is still usable, but corrective actions need to be performed.	Immediate follow-up activity to determine the resolution and the corrective actions needed to be performed to avoid another erroneous termination in the future.	<ul style="list-style-type: none"> ■ Error messages
Fatal	The system or a subcomponent cannot recover from error, and the severe situation causes fatal termination. The system or the software application is not usable anymore and cannot be started without corrective actions by experts.	Immediate follow-up activity to analyze the logs and traces in order to determine further investigations before being able to identify the correction actions to be performed.	<ul style="list-style-type: none"> ■ Error messages

Two log severity thresholds define this filtering, depending on two log categories (application or system). The increasing order of the severity threshold levels are:

Log Severity Threshold	Description	Scope	Generated Logs by Severity
INFO (default and recommended)	The SAP CC Core Server system generates all the possible log messages for the IT administrators or IT consultants. It is the highest possible threshold.	This log severity threshold is recommended for a productive SAP CC system in your system landscape. It enables all the technical operations with your system: log monitoring, error troubleshooting, root cause analysis.	<ul style="list-style-type: none"> ■ Info ■ Warning ■ Error ■ Fatal
WARNING	The SAP CC system generates only the log messages associated to	This log severity threshold is not recommended as a lot of valuable logs	<ul style="list-style-type: none"> ■ Warning ■ Error ■ Fatal

	warnings and errors (including fatal).	are not recorded (event notification, status recording, context information).	
ERROR	The SAP CC system generates only the log messages associated to errors (including fatal).	This log severity threshold is not recommended.	<input type="checkbox"/> Error <input type="checkbox"/> Fatal
FATAL	The SAP CC system generates only the log messages associated to fatal errors.	This log severity threshold is not recommended.	<input type="checkbox"/> Fatal

 **NOTE**

The severity level threshold can be either modified immediately during runtime or deferred after a restart of the concerned instances. For further information, refer to the “Modifying the logging and tracing policy” procedure available in the [SAP Convergent Charging Tuning Guide](#) documentation.

Categories

SAP CC defines the following categories of logs to ease troubleshooting operations and identify the audience:

Log Category	Description	Audience	Subcategory
Applications	This category contains messages related to application operation tasks	Application administrator	<input type="checkbox"/> Archiving <input type="checkbox"/> Backup <input type="checkbox"/> Configuration <input type="checkbox"/> Failover <input type="checkbox"/> Infrastructure <input type="checkbox"/> Resources <input type="checkbox"/> Security
System	This category contains messages related to system operation tasks.	SAP CC system administrator	<input type="checkbox"/> System changes <input type="checkbox"/> System configuration <input type="checkbox"/> System database <input type="checkbox"/> Enterprise services <input type="checkbox"/> Network <input type="checkbox"/> Security <input type="checkbox"/> Server <input type="checkbox"/> User interface

 **NOTE**

You can configure SAP CC to split the logs between two different destinations, one for the application administrators and another one for the system administrators.

Tracing

Tracing is a process which consists in writing advanced or internal information about an operation into an output file. Trace records provide the following information dedicated to support specialists:

- A detailed sequence of statements which describe the events of an operation as they are executed by an instance in the SAP CC system
- Diagnosing of an abnormal condition



IMPORTANT NOTE

The trace generation impacts the global performances of the SAP CC systems and may lead to business errors (charging operations not replied on time, peaks of latencies and response times, throughput decrease).

As trace messages are used by support specialists for understanding the program execution path and the internal status of the application, the following types of trace messages can be distinguished:

Typical Trace	Description	Action	Trace Severity
Program flow information	Information about which part (subroutine/method) of the program is executed and what control flow action is performed (method entered or exited, normally or via exception)	Advanced troubleshooting: <ul style="list-style-type: none"> ■ Problem determination ■ Problem location 	Path
Program context information	<ul style="list-style-type: none"> ■ Information about the semantically execution step ■ Important return codes of calls (like accesses to external resources, resource bundles, processing engines, or libraries) 	Advanced troubleshooting: <ul style="list-style-type: none"> ■ Problem determination ■ Problem location 	Info
Program context information related to an error	<ul style="list-style-type: none"> ■ Results of condition checks (assertions). If there are critical conditions which need to be satisfied in specific places of the code (otherwise the processing has to be terminated), a trace message describing the failed condition must be issued. ■ Technical information in application termination conditions. Normally, as part of the error handling, an exception is created. A notification on this event is written as a log message. If it is necessary to write additional technical context information (like variable values, etc.), this is written as a trace with the ERROR severity. 	Advanced troubleshooting: <ul style="list-style-type: none"> ■ Problem determination ■ Problem location 	Error

Trace Severities, Threshold, and Filtering

An important part of any trace message is its severity. It denotes the level of importance or relevance of a certain message. Traces can be restricted to certain severity levels, that is, only data of a defined severity is generated and collected. A severity level threshold defines this restriction. The increasing order of the severity levels is:

Trace Severity	Description	Audience
DEBUG (lowest severity)	Information for debugging purpose only valuable for support specialist to analyze the internal status of a program that composes the SAP system. This trace severity is reserved for internal use only and should not be used for production landscapes.	Support specialist

PATH	Information for tracing the execution flow inside the system or a subcomponent (entering and leaving methods, looping and branching operations) to understand the processing. This mode is very important for tracing operations when troubleshooting an SAP system.	Support specialist (level 3)
INFO	Extra information for announcing what has been performed by an SAP system or a subcomponent to better understand and trace the business logic, e.g. business document numbers, status changes, and so on.	Support specialist
WARNING	The application can recover from an anomaly and fulfill the required task, but needs attention from a developer or operator.	Support specialist
ERROR (default)	Uncorrectable error conditions in program execution or technical context information. The SAP system can recover from an error, but it cannot fulfill the required task due to this error. The SAP system may handle an exception and provide related information.	Administrator or Support specialist
FATAL (highest)	The application cannot recover from an error, and the situation causes fatal termination.	Administrator or Support specialist

A trace severity threshold defines this filtering. The increasing order of the severity threshold levels are:

Trace Severity Threshold	Description	Scope	Generated Traces by Severity
DEBUG	SAP CC generates all the possible trace messages for the support specialist of SAP lab experts. It is the highest threshold level.	This highest severity threshold for debugging purpose by SAP labs, with extensive and low level information	<ul style="list-style-type: none"> ■ Debug ■ Path ■ Info ■ Warning ■ Error ■ Fatal
PATH	The SAP CC system generates all the trace messages, except the debug traces that are dedicated to SAP labs or L3 support specialists.	For tracing the execution flow in the system (or Java Virtual Machine environment), e.g. used in the context of entering and leaving a method, looping, and branching operations	<ul style="list-style-type: none"> ■ Path ■ Info ■ Warning ■ Error ■ Fatal
INFO	The SAP CC system generates: <ul style="list-style-type: none"> ■ Trace messages associated to warnings and errors (incl. fatal) ■ Informational text, mostly for echoing what has been performed 		<ul style="list-style-type: none"> ■ Info ■ Warning ■ Error ■ Fatal
WARNING	The SAP CC system generates only the trace messages associated to warnings and errors (incl. fatal).	This severity threshold is interesting for a test, quality assurance, or pre-production SAP CC system during the implementation phase of the	<ul style="list-style-type: none"> ■ Warning ■ Error ■ Fatal

		integration project or during the operation phase. SAP CC application can recover from anomaly, and fulfill the desired task, but need attention from developer/IT integrator/IT administrator/operations team/support specialist.	
ERROR (default and recommended)	The SAP CC system generates only the trace messages associated to errors (incl. fatal). It is the default and recommended threshold in a productive SAP CC system in your system landscape.	This trace severity threshold is recommended for a productive SAP CC system.	<input type="checkbox"/> Error <input checked="" type="checkbox"/> Fatal
FATAL	The SAP CC system generates only the trace messages associated to fatal errors. SAP CC application cannot recover from error, and the severe situation causes fatal termination.	This trace severity threshold is not recommended. The SAP CC system generates only fatal error traces.	<input checked="" type="checkbox"/> Fatal

**NOTE**

The severity level threshold can be either modified immediately during runtime or deferred after a restart of the concerned instances. For further information, refer to the “Modifying the logging and tracing policy” procedure available in the [SAP Convergent Charging Tuning Guide](#) documentation.

Tracing Domains

For support specialists, SAP Convergent Charging defines some tracing domains which can be used to filter or extend the recorded trace messages.

For further information and recommendation about these domains, refer to the description of the LS_TRC_DOMAIN parameter available in the [SAP Convergent Charging System Parameters Reference](#) documentation.

**EXAMPLE**

You might need that your SAP CC Core Server system (or a given SAP CC instance) generates additional trace messages belonging to the SQL or JCO tracing domains.

Performance Traces

SAP Convergent Charging can generate performance traces to facilitate the troubleshooting of isolated performance issues. These performance traces represent specific trace messages that are recorded in the standard trace files of the SAP CC Core Server system. Collected information includes the execution path and processing time relating to a particular process step.

These trace messages:

- Belong to the PERF tracing domain

- Provide your SAP Support Team with new tracing tools that can be used to troubleshoot performance issues relating to isolated symptoms. For further information, refer to the [SAP Convergent Charging Operations Guide](#) documentation

Targeted Tracing

The Core Server system can activate and adjust the log and trace generation, automatically and temporarily. This targeted tracing only concerns the log and trace events relating to the processing of a particular operation request including all the subsequent activities.

By default, each instance of the Core Server system manages the following local elevations of severity thresholds:

- The log severity thresholds increase to INFO that is the default and highest level (recommended in a productive system)
- The trace severity threshold increases to WARNING. Depending on the configuration, it can increase to INFO, PATH, or DEBUG (highest level)

This feature is supported by the following operations:

Technical Interface	Availability	Description	Use
Web Services	All operations	SAP CC receives a custom SAP Passport from an interfaced system. The included trace flags configure the behavior of SAP CC (threshold levels, tracing domains).	End-to-end or process tracing
Message TCP and Web Services	<ul style="list-style-type: none"> ■ Immediate charging ■ Blank charging ■ Session-based charging 	During a tracing session, SAP CC equips a selected charging operation with a predefined passport.	Process tracing

Targeted tracing provides your SAP Support Team with new tracing tools that can be used to troubleshoot performance issues relating to isolated symptoms. For further information, refer to the [SAP Convergent Charging Operations Guide](#) documentation, or to the [SAP Passport Handling](#) section afterwards.



IMPORTANT NOTE

The targeted tracing triggers the trace generation. This may thus impact the global performances of your SAP CC landscape and may lead to business errors (timeouts during charging operations, peaks of latencies and response times, throughput decrease, and so on). Such business errors relate to the operations in the scope of the active targeted tracing but also to all other business operations in progress.

Output Formats

Each output destination uses a format for both logs and traces:

- List (default format for the logs of SAP CC)
- Trace
- XML
- Highdeal (for compatibility reasons)



NOTE

In SAP CC 5.0, you cannot change the format of the logs of the BART Server system.

List Format

The List format is a standard human readable format which can be parsed and which is used by SAP Log Viewer. The values of each element are written into files using a hash separator (#). SAP CC does not fill in each value and several hashes may be grouped in a record.



EXAMPLE

A dispatcher instance records the following log message in the dispatcher#1_ldf1.0.log file:

```
#2.0#2013 02 20 08:55:48:015#+0100#Info#/Applications/Common#com.sap.SCC.core_server.000000#IS-
CC##C0000A48C82A0000000000002AF6A882####<system>#####Thread[main,5,main]#Plain##Starting an
instance for SAP Convergent Charging 4.0 (Technical Version Number: 4.5.0.0)#
...
#2.0#2013 02 20 08:55:56:928#+0100#Info#/Applications/Common#com.sap.SCC.core_server.000026#IS-
CC##C0000A48C82A000000000000A2AF6A882####<system>#####Thread[main,5,main]#Plain##Master
Dispatcher Internal Service started and available at the address: 10.68.24.42:2100#
```

This message relates to the “com.sap.SCC.core_server.000000” unique message identifier. For further information about this message, refer to the [SAP Convergent Charging Log Message Reference](#) documentation.

Trace Format

The trace format is a human readable text format with a simplified structure suitable for troubleshooting operations:

```
<Timestamp in a readable format> - <Location on behalf of which the message has been
created> [<Thread that has emitted the message>] - <Severity of the message> - [Log
code in CC 3.0] - <Message short text>
```

The content of such messages is a selection of the main information items which are relevant for troubleshooting operations.



EXAMPLE OF A LOG MESSAGE

A Dispatcher records the following log message in the dispatcher#1_ldf1.0.log file:

```
2013-02-20 08:55:48.015 - [10.68.24.42]dispatcher#1 - INFORM - [LAUNCHER_STARTING] - Starting an
instance for SAP Convergent Charging 4.0 (Technical Version Number: 4.5.0.0)
```

This message relates to the “com.sap.SCC.core_server.000000” unique message identifier. For further information about this message, refer to the [SAP Convergent Charging Log Messages Reference](#) documentation.



EXAMPLE OF A TRACE MESSAGE

An Updater records the following trace message in the updater#1_LDF2.0.log file:

```
Mar 2, 2013 12:03:30 AM com.highdeal.pnr.hci.server.SearchExportableItemTemplateOpImpl
[Thread[http-srv-3,5,main]] Warning: Unable to search exportable item template: SAP CC Server
cannot connect to SAP CI system
```

Where:

- **Date/Time** is: Mar 2, 2013 12:03:30 AM
- **Path** is: com.highdeal.pnr.hci.server.SearchExportableItemTemplateOpImpl [Thread[http-srv-3,5,main]]
- **Severity and Message** are: Warning : Unable to search exportable item template: SAP CC Server cannot connect to SAP CI system

XML Format

This format is an XML-based format suitable for file transfer to be further processed in other applications.



EXAMPLE OF AN XML LOG MESSAGE

A Taxer records the following log message (Message ID: com.sap.SCC.core_server.000000):

```
<record>
  <id>C0000A48C82A0000000000000000E90E23</id>
  <time>1361875299235</time>
  <source>/Applications/Common</source>
  <application></application>
  <location></location>
  <user>&lt;system&gt;</user>
  <session></session>
  <transaction></transaction>
  <dsr-component></dsr-component>
  <dsr-user></dsr-user>
  <dsr-transaction></dsr-transaction>
  <thread>Thread[main,5,main]</thread>
  <severity>Info</severity>
  <relatives>
    <relative></relative>
  </relatives>
  <msg-type>Java</msg-type>
  <msg-code>com.sap.SCC.core_server.000000</msg-code>
  <msg-clear>Starting an instance for {0} (Technical Version Number: {1})</msg-clear>
  <args>
    <arg>SAP Convergent Charging 4.0</arg>
    <arg>4.5.0.0</arg>
    <arg>LAUNCHER_STARTING</arg>
  </args>
</record>
```

Output Destinations

SAP CC manages two main output destinations for logs and traces:

- The console, which can be used to print logs and traces
- Rotating files, which represent the default output destination. SAP CC gives the possibility to define a logging policy where you can configure up to 5 output channels and define the content of each channel (number of files, maximum size, file naming and location, output format). Every channel is represented by a fileset, and a rotation occurs when the maximum file size is reached.

Configuration

The logging and tracing policy of SAP CC relies on multiple configuration parameters which give the possibility to adapt the logged content to specific and/or temporary needs.

Depending on your system or application, you configure the logging and tracing functions by setting up the:

- System parameters of the Core Server system

- Configuration files (BART Server, Diameter Server, IEC, Cockpit)
- Launch scripts

For further information about the customizing activities and configuration settings of SAP CC, refer to the [SAP Convergent Charging](#) documentation.

 **NOTE**

You can mix both logs and traces to the same destination for development landscapes to facilitate the troubleshooting operations (Core Server only).

SAP Passport Handling

SAP Convergent Charging supports the SAP Passport technology in order to provide your SAP Support Team with advanced tracing features and supportability tools that can be used for operations such as:

- End-to-end tracing
- Process tracing and business process tracing

The SAP Passport concept relies on a byte array structure which is included into the concerned requests and gives the possibility to:

- Improve and clarify the logged information
- Trace system activities in a distributed environment, in order to ease support and maintenance operations
- Determine the overall time and resource consumption, for statistical analysis purposes or performance optimization

 **NOTE**

SAP CC does not handle the SAP Passport technology to support the Single Sign-On (SSO) concept. Consult the [SAP Convergent Charging Security Guide](#) documentation for more information about SAP user administration and authentication.

Supported SAP Passports and Available Trace Flags

SAP Passports are made up with different properties, whose values are encoded using a byte array structure. SAP CC supports both the 2 and 3 versions of the SAP Passports, which contain the following properties:

Property	Version		Description
	2	3	
Eye catcher 1	■	■	Hard coded value used to determine the beginning of the passport
Version	■	■	Version of the SAP Passport. Versions 2 and 3 are supported by SAP CC.
Length	■	■	Length of the byte array structure
Trace flag	■	■	Trace flags represented as a byte, each bit of this byte being a trace flag that can trigger additional traces. SAP CC uses the trace flag to activate and adjust the targeted tracing, as described in the next table. Each trace flag causes additional trace output in the generated trace messages.

Initial system or component ID	■	■	System and component name. For a J2EE system DSR component, the format is: <host><SID><clusterId>. This field is generated by the system which initiates the request and is not be modified during the propagation of the SAP Passport
Initial service type	■	■	Integer value that corresponds to the type of the request (HTTP, RFC). This field is generated by the system which initiates the request and is not be modified during the propagation of the SAP Passport.
Initial user ID	■	■	User name. This field is generated by the system which initiates the request and should not be modified during the propagation of the SAP Passport.
Initial action	■	■	String representation of the executed action
Initial action type	■	■	Integer value which represents the type of the executed action. This is only relevant for HTTP actions and the possible types are: <ul style="list-style-type: none"> ■ Web container ■ Web service ■ EJB
Previous system or component ID	■	■	Name of the system (or component) where the request came from. The format is the same as the one used for the 'component name' property. The ID of the previous system (or previous component) is always set with a system or component's own ID before the passport is sent. This means that the direct predecessor can be identified when the passport is received.
Transaction ID/GUID	■	■	Unique identifier of the request
Connection ID		■	
Connection counter		■	
Eye catcher 2	■	■	Hard coded value used to determine the end of the passport

**NOTE**

The SAP Passport includes a trace flag property that defines the relevant trace flags. The SAP CC system uses this information to activate and adjust the targeted tracing. Customizing these trace flags enables you to configure the levels and categories of generated logs and traces.

The following table describes the behavior of SAP CC regarding value of the bit mask:

Bit	Trace Flag	Tracing Elevation
0	SQL Trace	Targeted domain: SQL When this bit is set, the traces with SQL domain are recorded by the SAP CC system.
1	Buffer Trace	Not used by SAP CC
2	Enqueue Trace	Not used by SAP CC
3	RFC Trace	Targeted domain: JCO
4	Auth. Trace	Not used by SAP CC
5	C function Trace	Targeted domains: all the domains different from the tracing domains that are already associated to one of the following trace flag: SQL, JCO, DATA, PERF, HCI, and WS When this bit is set, all the tracing domains that have no dedicated bit are recorded by the SAP CC system.

SAP Passport and End-to-End Tracing

In a multi-system landscape, a request can be propagated to different systems. When such a situation occurs, it can be necessary to identify error or trace messages and associate them to a particular operation request that was taken into account by the different concerned systems.

To provide such an end-to-end tracing feature, SAP Convergent Charging handles *SAP Passports* via the *Web Services* technical interface. All the available service operations support an SAP Passport.

Depending on the trace flag information that is available in the received passport, the SAP CC system activates or adjusts both the log and trace recording automatically and temporarily. This targeted tracing only concerns the log and trace events that relate to the processing of the flagged request including all the subsequent activities. The end-to-end tracing can trigger the process tracing. This combination enables the support specialists to consider particular treatments in the SAP CC system but also in third-party systems (database, provisioning, CRM).

The SAP CC system does not modify the received trace flags.

Each instance of the SAP CC system can perform the default and custom targeted tracing function:

Targeted Tracing	Log Severity Thresholds	Trace Severity Threshold	Generated Traces
Default	The log severity thresholds increase to INFO that is the default and highest level. This level is recommended in a productive system.	The trace severity threshold increases to WARNING locally, except the traces that have a tracing domain defined.	Depends on received custom SAP Passports
Configuration and Customizing	N/A	In your front-end systems or test system, you can configure the transmitted SAP Passport to increase the trace severity threshold locally to: <ul style="list-style-type: none"> ■ WARNING ■ INFO ■ PATH ■ DEBUG (highest level) 	In your front-end systems or test system, you can configure the transmitted SAP Passport to generate the trace messages that relate to the following tracing domains: <ul style="list-style-type: none"> ■ SQL domain ■ JCo domain ■ HCI and WS domains ■ PERF domain ■ DATA domain ■ All other domains defined in SAP CC



NOTES

- SAP CC manages the propagation of a passport between all the relevant instances of a SAP CC system
- SAP CC does not manage the propagation of a passport to asynchronous and induced requests to external systems
- For communications made upon the SOAP over HTTP communication channel, the SAP Passports are included in the header of the HTTP messages, using a "SAP-PASSPORT" property. For further information about the communications channels, refer to the [SAP Convergent Charging Security Guide](#) documentation and to the [Web Services Documentation](#)

- Consider the prerequisites and limitations of the targeted tracing function by referring to the notes about this function in the [Enhanced Logging and Tracing Framework](#) section

SAP Passport and Business Process Tracing

If isolated irregularities occur during SAP CC operations, it might be necessary to identify the execution error or execution path available in trace messages, and associate them to a particular operation request leading to a performance issue. This operation is associated to a provider contract (or a subscription) that contains the charging logic (pricing, allowance) and end-customer business data to analyze. To be complete, the analysis must concern the end-customer business data that is configured in the subscriber account associated to the contract.

To provide such a process tracing feature, SAP Convergent Charging can generate custom *SAP Passports* automatically and equip the next received charging operation request that relates to the selected subscriber account. The service operation can be executed using the Message TCP and the Web Services technical interfaces.

This function is supported by the following operations:

Service operation	Message TCP			Web Services
	Online Charging Service	Offline Charging Service	Stateless Rating Service	Charging Service
Event-Based Charging				
Immediate charging	■	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Single charging	<input type="checkbox"/>	■	<input type="checkbox"/>	■
Blank charging	■	■	<input type="checkbox"/>	■
Batch charging	<input type="checkbox"/>	■	<input type="checkbox"/>	■
Immediate rating	<input type="checkbox"/>	<input type="checkbox"/>	■	<input type="checkbox"/>
Session-Based Charging				
Start, stop, and update operations	■	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

- Yes
- No
- N/A

In Admin+, you can arm the targeted tracing function for the next incoming charging operation request that relates to a specified subscriber account. The SAP CC instance responsible for the specified subscriber account activates or adjusts the recording of both log and trace, automatically and temporarily. It generates all the trace messages belonging to the following tracing domains:

- DATA
- PERF (performance traces)

EXAMPLE

By default, the configured targeted tracing records the trace messages that belong to the DATA domain. You can capture the rating context information during a charging operation.

IMPORTANT NOTES

- This targeted tracing only concerns the log and trace events that relate to the processing of the flagged request including all the induced activities. The SAP Passport is active for the thread that processes the request until the request ends and is thus also used for other processing activities in the same thread, such as the other

requests without any timeout configured in the stateful service client implemented in your mediation system or client application

- The recording of traces impacts the global performances of the SAP CC system and may lead to business errors (charging operations not replied on time, response time increase, throughput decrease). These business errors relate both to the operations in the scope of the active targeted tracing but also to all other business operations in progress
- Consider the prerequisites and limitations of the targeted tracing function by referring to the notes about this function in the [Enhanced Logging and Tracing Framework](#) section

Logging and Tracing Framework of Diameter Server

The Diameter Server uses the Java default logging framework associated to dedicated configuration parameters. The Diameter Server generates multiple rotating log files.

Output Format

The output format is a human readable text format with a simplified structure suitable for troubleshooting operations:

```
<Timestamp in a readable format> <Location on behalf of which the message has been
created> <Thread that has emitted the message> <Severity of the message>: <Message
short text>
```

The content of such messages is a selection of the main information items which are relevant for troubleshooting operations.



EXAMPLE

The unique instance of a Diameter Server system records the following log messages:

```
23 mai 2013 14:29:50 com.highdeal.diameter.server.Launcher main
INFO: Starting SAP Convergent Charging 3.0 Diameter 4.4.11.0.
23 mai 2013 14:29:50 com.highdeal.diameter.server.Launcher main
INFO: SAP SE or an SAP affiliate company 1.6.0_43 platform on Windows 7 6.1 amd64.
23 mai 2013 14:29:51 com.highdeal.diameter.server.Launcher main
INFO: Charging stack started with host: <[10.31.104.59]> and port: <[2000]>.
23 mai 2013 14:29:52 com.traffix.openblox.core.utils.logging.SimpleLoggingHandler doLog
CONFIG: Initiating Configuration from D:\usr\sap\D31\CAD11\config\serverConfig.xml
23 mai 2013 14:29:52 com.traffix.openblox.core.utils.logging.SimpleLoggingHandler doLog
CONFIG: SETTING_ROOT_LOGGER_LEVEL INFO
23 mai 2013 14:29:52 com.traffix.openblox.core.utils.logging.SimpleLoggingHandler doLog
INFO: LOADING_STANDARD Ro 3GPP TS 32.299 V11.1.0
23 mai 2013 14:29:52 com.highdeal.diameter.server.Launcher main
INFO: SAP Convergent Charging 3.0 Diameter 4.4.11.0 started.
```

Configuration

For further information about the customizing activities and configuration settings related to the `logging.properties` file, refer to the [SAP Convergent Charging documentation](#).

User Session Management

In Desktop Applications

For security reasons, SAP Convergent Charging provides a technical feature which enables you to monitor and control the use of the Core Tool and BART Tool user interfaces. This User Session Management feature gives the possibility to:

- Control the number of user sessions that can be opened in parallel on a given user interface for a given user
- Specify a maximum duration for interactions between the user interfaces and the servers they are connected to
- Specify a maximum duration for inactivity periods (which corresponds to periods where no interaction occurs between the user interfaces and the servers they are connected to)
- Display the list of opened user sessions, using the "search_user_session" command of Admin+
- Remotely terminate a given user session, using the "delete_user_session" command of Admin+



NOTES

- The User Session Management feature relies on a framework which can be implemented for a homemade user interface connected to the Core Server system
- For further information about the configuration of this feature, refer to the [SAP Convergent Charging Configuration Guide](#) and [SAP Convergent Charging Parameters Reference](#) documentations
- For further information about the commands available for listing and deleting user sessions, refer to the "Working with Admin+" section of the [SAP Convergent Charging User Interfaces](#) documentation

User Session Lifecycle

Maintained by the Core Server, every user session:

- Starts when a human user logs on the Core Tool or the BART Tool user interfaces
- Is considered as active as long as the duration between two server/user interface interactions is shorter than the duration configured in the USER_SESSION_VALIDITY_PERIOD parameter
- Ends:
 - Automatically if the human user closes the user interface
 - Automatically if the user session remains inactive for a period which is longer than the configured validity period
 - Automatically when the communication with the server (that the user interface is connected to) fails, e.g. in case of updater failure or restart
 - Manually when the administrator remotely terminates it



NOTE

If a user session ends while a human user is still connected to the user interface, a notification is sent to the user to inform that a re-authentication is necessary, in order to start a new user session.

In Web Application

For security reasons, SAP Convergent Charging provides a technical feature which enables you to control the use of the Cockpit user interface. This User Session Management feature gives the possibility to:

- Lock user sessions manually or automatically after an inactive period. When this period is reached, the SAP user must log on again to continue

- End the working sessions automatically after an inactive period. All unsaved modifications are lost
- Alert power users about the session expiration and potential risks



NOTES

- The User Session Management feature for the Cockpit web application relies on another framework that is not the framework used for the desktop applications (Core Tool, BART Tool). As a consequence, you cannot monitor the use in Admin+
- For more information about the configuration of this feature, refer to the [SAP Convergent Charging Configuration Guide](#) documentation.
- For further information about the commands available for listing and deleting user sessions, refer to the "Working with Admin+" section of the [SAP Convergent Charging User Interfaces](#) documentation

Table rollover for rating sessions

As described in the description of the “[Dual Database](#)” feature, the Session database is an optional database that contains the different data related to the business objects handled by SAP CC during the session-based charging operations. During such operations, different kind of events occur, each event impacting the database table that is used to store the different information related to the rating session.

Whatever the impact is (new record, update of an existing record, or record removal), the quantity of modifications performed on this database table may lead to its fragmentation, that decreases performances as time goes by.

To reduce this fragmentation, SAP Convergent Charging provides a rollover mechanism that gives the possibility to store the new created charging sessions among a set of tables instead of a single one. Enabling this rollover mechanism leads to the switching of the charging sessions data among up to 4 destination tables, according to a defined period of time. When this period is over, the destination table changes and the previously used destination starts to empty according to the end of the pending sessions that are associated to this destination table.

This mechanism relies on the following 2 system parameters, that you can configure to fit specific needs:

- **RATING_SESSION_TABLE_ROLLOVER_COUNT**, used to specify the number of rating session tables that can be used as destinations
- **RATING_SESSION_TABLE_ROLLOVER_PERIOD**, that defines the validity period (in days) of the table that is currently used as destination

For further information about the configuration of this rollover mechanism, refer to the "Modifying the table rollover policy for charging sessions" section available in the [SAP Convergent Charging Tuning Guide](#) documentation.

License Management

SAP Convergent Charging provides a License Management feature to support the following technical functions:

- Temporary SAP licenses
- Permanent SAP licenses
- SAP license auditing

Temporary SAP License

You must install a permanent SAP license. When you install your SAP Convergent Charging landscape, a temporary license is automatically installed.

Permanent SAP License

Before the temporary license expires, you must apply for a permanent license key from SAP. For further information, refer to the “Installing a permanent license” procedure available in the [SAP Convergent Charging Installation Guide](#) documentation.

For additional information about SAP license keys and how to obtain them, see <http://support.sap.com/licensekey>.

SAP License Auditing

SAP Convergent Charging provides some tools to facilitate the SAP license audit as an SAP System Measurement. The SAP CC system measurement is based on usage metrics that are recorded daily. Only the yearly peak values are incorporated in the licensing information.

SAP Global License Auditing Services supports all SAP customers in fulfilling their contractual duty to carry out system measurements.

Deployment Scenario	SAP Program Measurement	Measurement ID and Engine
Scenarios with SAP Solutions (S4/HANA, SAP Business Suite)	Engine Measurement	Measurement ID: “0590” Engine: “ SAP Convergent Charging ”
Standalone scenario	Self-Declaration Product Measurement	

To create your Self-Declaration Product Measurement:

- For each productive Core Server system, you prepare some Global License Auditing Services (GLAS) reports with the Cockpit and Core Tool user interfaces
- You verify that the measured values are compliant with the system sizing and the estimated data growth and traffic load increase
- Ensure that you have adequate licensing for your installation and plan some extension
- When you are ready to share the auditing results, send the report file to your SAP license auditor or consider the self-declaration on SAP Support Portal

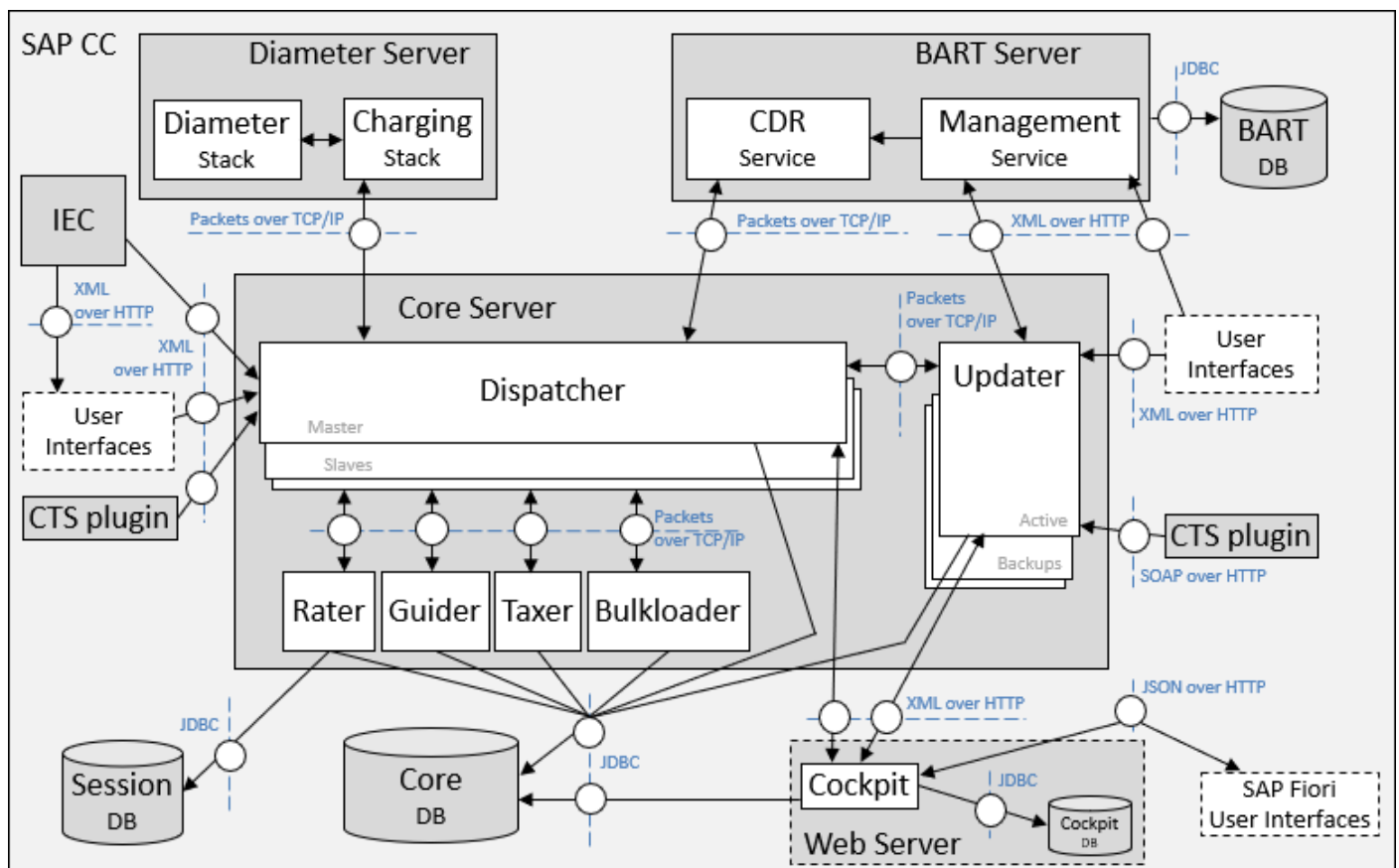
In this documentation, refer to the “Product Usage Measurement” section in the “System and Data Auditing” chapter for more information about the concepts about the product usage metrics, the administrative tools, and the GLAS file format. For additional information about SAP System Measurement, see <https://support.sap.com/en/my-support/systems-installations/system-measurement/system-measurement-information.html>.

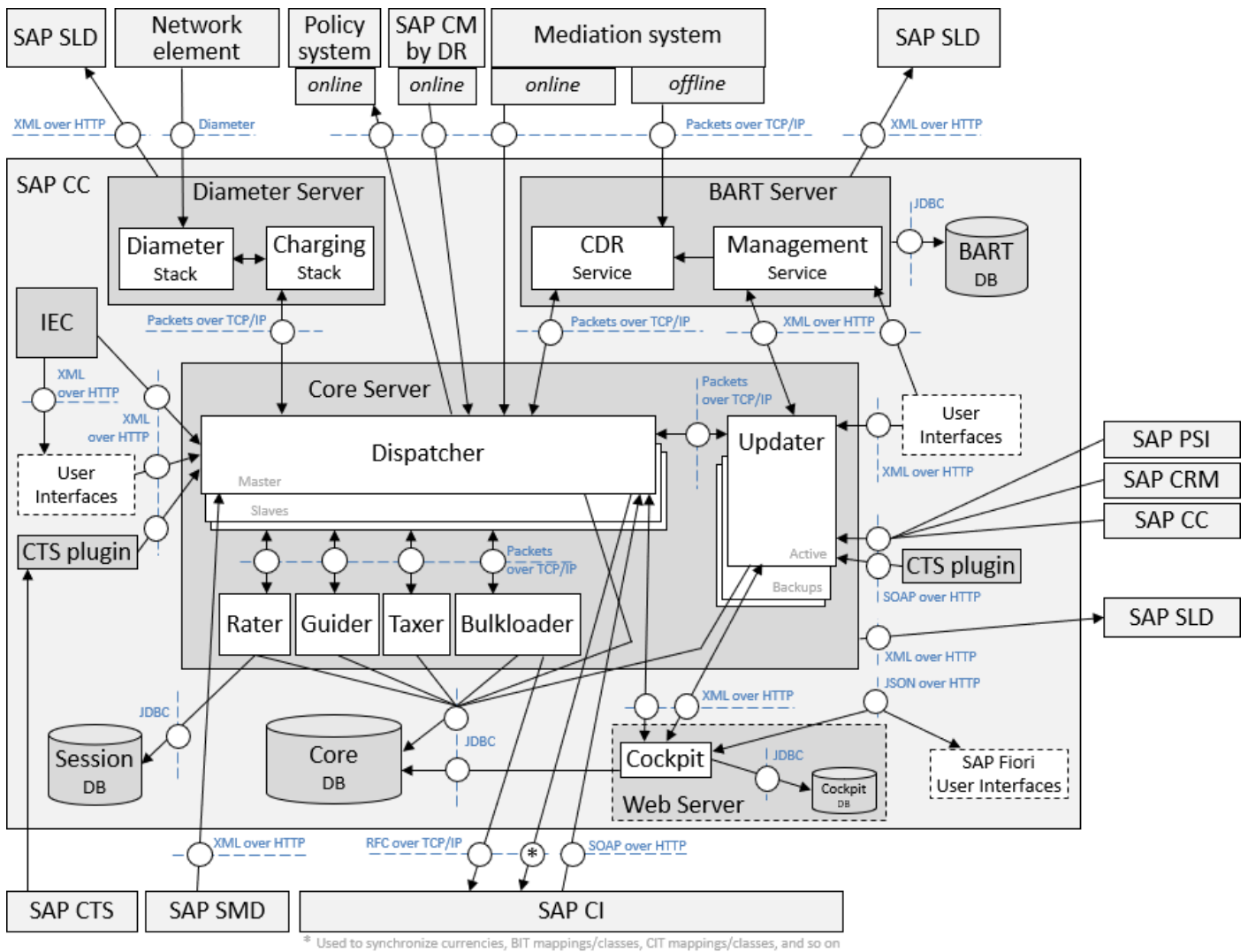
5 Architecture

Synthetic Overview

The 2 following schemas provide a synthetic view of the SAP CC architecture. The first schema only focuses on SAP CC, whereas the second schema includes the different interactions with other elements such as:

- A network element, connected with the Diameter Server
- A mediation system, connected either with:
 - The master Dispatcher of the Core Server, for online mediation purposes
 - The BART Server, for offline mediation purposes
- SAP CRM, SAP CI and SAP CTS, in case of an integrated *SAP Solution* scenario with the SAP ERP/FI-CA, SAP CRM and SAP CTS components of SAP Business Suite





SAP Convergent Charging contains elements and objects that can be administered and managed through user-oriented tools considered as client applications in the above schema. According to the needs, these tools can be:

- Console applications, mainly used for configuration and administration of the system and data
- Desktop applications, dedicated to business purposes and providing user-friendly graphical interfaces
- Web applications in a central and modern launchpad

The table below shows the list of provided tools, which are detailed in the documentation of the concerned components:

Tool	Core Server	BART Server	Diameter Server	IEC
Console applications				
> Setup Tool	■			
> Config Tool	■			
> Admin+	■			
> HTTP Client	■			
> BART+		■		
> BART Setup Tool		■		

Desktop applications				
> Core Tool	■			
> BART Tool		■		
> CAT Tool				■
Web applications				
> Cockpit	■			
> Analyze Item Files	■			
> Display System Status	■			
> Display Usage Metrics	■			
> Manage Mapping Tables	■			
> Manage Mapping Table Classes	■			
> Manage SAP CC System Parameters	■			

Components and Tools

Core Server

Made up with two functional modules (Rating Core and Balance Management), the Core Server component represents the rating engine of SAP CC. This component is in charge of:

- Defining the pricing, charging and refilling algorithms and structures (grouped by catalogs associated to a service provider)
- Computing prices which correspond to services consumptions (taking into account the associated taxation requirements)
- Defining the accounts to charge according to the defined charging configurations

The Core Server implements the following general processes:

- Chargeable Items Charging, which is used to:
 - Rate external and internal events
 - Charge subscriber accounts balances which are related to prepaid accounts or postpaid accounts
- Session-based Charging, which represents an execution mode of the Chargeable Items Charging process with additional features such as credit-control, credit reservation, reservation cancellation, reservation renewal or aggregation
- Prepaid Accounts Refilling, which is used to refill balances related to prepaid accounts
- Activation, which gives the possibility to generate internal events when an external event is received
- Product Modeling, which is used to manage offers or charge plans and associated business objects
- Order Management, which is used to manage subscriptions, provider contracts and accesses
- Tax Calculation, which is used to compute real-time EU VAT and US Telco taxes

The Core Server is delivered with a set of user interface applications which give the possibility to:

- Administer its behavior (from installation to configuration and operating steps)
- Manage the handled master data

The following list provides some detailed information related to these applications:

Core Tool

Core Tool is a desktop application written entirely in Java. The GUI of this application simplifies the management of a large number of processes or operations available in SAP CC such as:

- New services and pricing structures implementation and customization
- Agreement management (provider contracts, subscriptions)
- Dynamic pricing and promotional launching
- Real-time customized offers and charge plans management
- Pricing and provisioning auditing

**NOTE**

For further information about Core Tool, refer to its dedicated [SAP Convergent Charging Online Help](#) documentation.

The Core Tool implements a technical mechanism named “Concurrent Edition Management”, that is used to prevent concurrent modifications on catalog objects. Relying on the [User Session Management](#) feature, this mechanism:

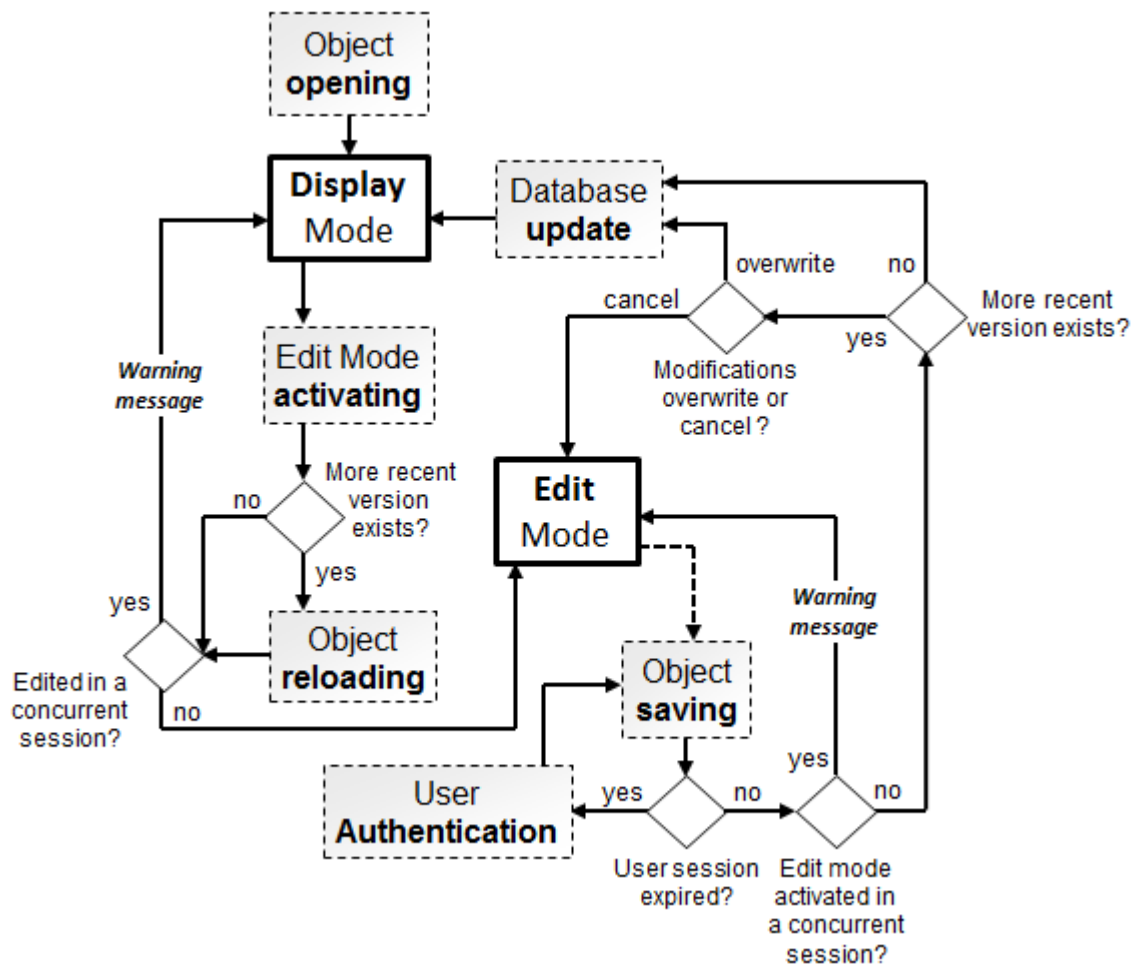
- Introduces a new “Display mode”, which corresponds to a read-only mode, and avoids users to perform modification on an object
- Introduces a new “Edit mode”:
 - That users must activate in order to modify an object
 - That cannot be used twice for a given object by forbidding: a user who tries to modify an object that is already opened for modifications in another user session cannot activate the “Edit mode” (and is automatically warned about the situation regarding the concurrent user and associated session)
 - That cannot be automatically re-activated after a user session expiration, in order to deal with possible parallel modifications

**NOTE**

If an object has been modified while a user session has expired, the user who tries to modify this object is notified about the concurrent performed modifications, and is proposed to:

- Overwrite the last changes with his/her modifications
- Cancel the operation, in order to save his/her modifications in a file for later corrections

The following schema summarizes the possible interactions between the 2 introduced modes:



 **CAUTION**

- The “import” operation of Core Tool overwrites any object, even if this object is opened by someone else in “Edit mode”
- The “Concurrent Edition Management” mechanism is only relevant for modifications performed by human users within Core Tool. As a consequence, this mechanism does not concern modifications performed by technical users via XML over HTTP (HCI) or SOAP over HTTP (Web Services) communications channels (e.g. SAP CRM which modifies mapping tables using Web Services)

Setup Tool

The Setup Tool is a console application dedicated to the management of the Core Server instances. It gives the possibility to manage:

- The licenses for all deployed instances
- The security settings required to secure communication channels (such as key pairs for encryption mechanisms)
- The notification policy
- The CIF configuration for Raters and Bulkloaders
- The RIF configuration for Raters
- The configuration related to the SAP ERP reference system for the Updaters

- The VAT taxation policies (VAT rate, VAT rules, country tax policies)
- The EZTax transaction/service types
- The list of transport destinations used for catalog transport purposes

**NOTES**

For further information about the Setup Tool and associated commands, refer to the [SAP Convergent Charging Configuration Guide](#) and [SAP Convergent Charging User Interfaces](#) documentations.

Config Tool

Config Tool is a console application dedicated to the configuration of the Core Server instances. It gives the possibility to:

- Export the persistent values of all the configuration parameters of the Core Server, which may be useful to transport a complex configuration of Core Tool from one landscape to another
- Import (and thus update) some configuration parameters into the Core database

Note that the import operation performs modifications on the Core database which are not immediately taken into account by some instances. As a consequence, it might be necessary to restart such instances. If you want to change the value of a single parameter or if you want to temporarily apply your modifications, you can use Cockpit or the “set” command of Admin+ described below.

**NOTES**

For further information about the Config Tool and associated commands, refer to the [SAP Convergent Charging Configuration Guide](#) and [SAP Convergent Charging User Interfaces](#) documentations.

Cockpit

Cockpit is a web application that gives the possibility for pricing specialists and system administrators to configure pricing data and manage the Core Server system. It provides power users with a graphical user interface to:

- Have a central place to administer SAP Convergent Charging with apps from your browser
- Configure and maintain the pricing data and configuration for the end-customer services
- Manage the persistent values and runtime values of the system parameters in the connected Core Server system
- Filter and visualize the recorded metadata related to the data files generated by SAP CC during the execution of the different processes

Cockpit represents a launchpad that contains apps based on SAP Fiori, the new user experience (UX) for SAP software that applies modern design principles.

As of SP 3, Cockpit includes the following apps, that both gives the possibility to manage all the mapping tables and mapping table classes that are necessary to build the pricing configuration of your marketable end-customer services:

- “Manage Mapping Tables”
- “Manage Mapping Table Classes”

As of SP 2, Cockpit includes the following apps:

- “Display System Status”, that gives the SAP CC administrators the possibility to quickly verify the system status and availability
- “Display Usage Metrics”, that gives the SAP CC administrators the possibility to verify and audit the global usage of system with respect to the licensing and sizing

As of SP 1, Cockpit includes the following apps:

- “Manage SAP CC System Parameters”, that gives to the SAP CC administrators the possibility to inspect and configure parameter values quickly and efficiently
- “Analyze Item Files”, that gives the possibility to inspect and supervise the different states of the data files generated by SAP CC

 **NOTE**

Cockpit is an optional user interface deployed in a Java Web server such as Apache Tomcat Server. It consists of a front-end component (the visible user interface) and a back-end component (OData Services). For more information about Cockpit and its applications, refer to its [primary help](#).

Admin+

Admin+ is a console application dedicated to the management of the Core server. It gives the possibility to establish a connection with a specific instance and modify both its running configuration and its configuration parameters. Through commands execution taking XML messages into account, this application performs operations such as:

- Listing running instances
- Getting or setting the values of instances’ parameters
- Managing batch rating groups
- Updating the log levels of instances
- Modifying messages output (console/file)
- Displaying service time statistics for one or multiple SAP CC charging client(s), used for [monitoring and/or troubleshooting purposes](#)

 **NOTE**

For further information about Admin+ and associated commands, refer to the [SAP Convergent Charging Configuration Guide](#) and [SAP Convergent Charging User Interfaces](#) documentations.

HTTP Client

HTTP Client is a console application used to send XML requests to the Core server. These requests correspond to XML fragments accepted by available APIs, and thus give the possibility to interact with a large number of business objects. This application behaves as a simple messages sender, and no control is made on sent messages and performed actions.

 **NOTE**

For further information about the HTTP Client tool and associated commands, refer to the [SAP Convergent Charging Solution Operations Guide](#) and [SAP Convergent Charging User Interfaces](#) documentations.



CAUTION

No control is made on XML sent messages. Considering that some operations can lead to database changes and thus impact data integrity, **it is highly recommended to use this application carefully.**

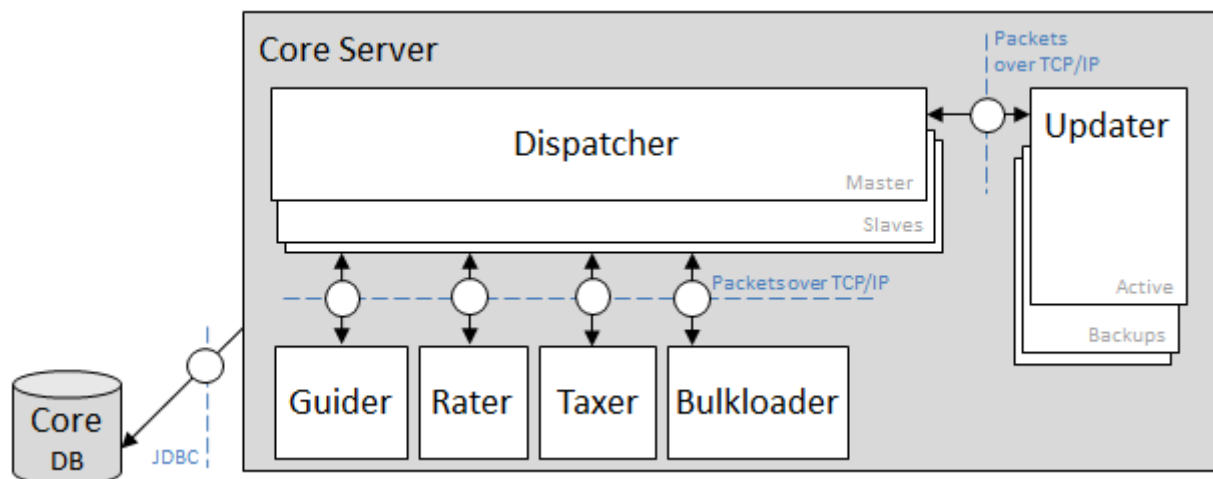
The architecture of the Core Server relies on a modular architecture which is built upon a set of instances acting as business services:

- **Dispatchers**, used to distribute the clients' requests to the adequate instance
- **Updaters**, used to manage business objects and provide up-to-date information to other instances
- **Guiders**, dedicated to guiding operations during charging services
- **Raters**, which perform the charging, recharging and refilling operations (and compute VAT taxes)
- **Taxers**, dedicated to real-time calculation of United States Telco taxes
- **Bulkloaders**, used to load charged item or refill record data files in bulk mode into SAP CI



NOTE

The deployed instances of the Core Server communicate together through the Packets over TCP/IP communication channel, which used to transport proprietary messages. For further information about this communication channel, refer to the [SAP Convergent Charging Security Guide](#) documentation.



The following table provides some detailed information related to these instances:

Dispatcher

- Master/slaves
- Caches
- RRL Map

The Dispatcher is a **public** instance in charge of distributing all incoming requests to the other adequate instances, and thus represents the central point of the Core Server. This instance can:

- Communicate with other instances using the Packets over TCP/IP communication channel

■ SSRL Map

- Communicate with the SAP CC CTS plugin using the XML over HTTP communication channel
- Be contacted by client applications (such as the Core Tool user interface) using the XML over HTTP or TCP/IP Packets over TCP/IP communication channels
- Be discovered by the other instances or client applications using the Messages over UDP communication channel

As Dispatchers have been implemented to handle and route all incoming requests, they are the only instances who know the up-to-date content of the dynamic instance map and thus the list of running instances. As a consequence, all the other starting instances have to contact any available Dispatcher. If the contacted Dispatcher is a slave Dispatcher, the request is automatically rerouted to the master one which is in charge of updating and reorganizing this dynamic instance map.

Master/slaves dispatchers

As described above, the static instance map contains a list of Dispatchers which are all deployed, started and coexisting in the running environment. All these Dispatchers know the list and state of each other running Dispatcher (described in the dynamic instance map), and handle the incoming requests. One of these Dispatchers is considered as the “master” Dispatcher, and all the other running Dispatchers are then considered as “slaves” Dispatchers.



NOTES

- Distinction between master and slave modes is made on startup according to the defined list of Dispatchers. The master Dispatcher corresponds to the first element of this list whose startup succeeded. In case of startup or running failure:
 - The master mode is transferred to the first available slave Dispatcher
 - The failed Dispatcher is retrograded to the last position in the list
- Incoming administration requests are only handled by the master Dispatcher. If an administration request is sent to a slave Dispatcher, an implicit redirection to the master Dispatcher is performed

Managed caches

Due to the fact that Dispatchers provide traffic handling mechanisms, they need to maintain in memory:

- An up-to-date list of users, for authentication purposes
- The partition mapping in order to guide a request to the adequate Guider and Rater

Reservation renewal listener (RRL) map

To provide the reservation renewal mechanism during session-based charging operations, the Dispatchers manage a map of client applications or systems which previously registered a listener (RRL) for reservation renewal purpose. Automatic re-registrations give the possibility to automatically update this map in case of Dispatcher failure or modification of the instance map.

Spending status report listener (SSRL) map

To provide a spending status reporting mechanism during online charging and refilling operations, the Dispatchers manage a map of client applications or systems which previously registered a listener (SSRL) for spending status monitoring purpose. Automatic re-registrations

give the possibility to automatically update this map in case of Dispatcher failure or modification of the instance map.

Guider

■ Caches

A Guider is an instance contacted only by the Dispatchers to get routing information related to a given access. Guiders cannot be contacted by a client application, and are thus considered as **private** instances, used for internal operations only.

Guiders have been implemented to:

- Collect on startup all the required information about accesses
- Build a service dictionary and a memory map containing subscriptions and users identifiers

Managed caches

Because Guiders mostly provide routing mechanisms, they need to maintain an up-to-date list of existing accesses. This cached structure is stored in memory, and uses a LRU mechanism in order to reduce memory consumption.



NOTE

For further information about the accesses cache and its associated LRU mechanism, refer to the [adequate section](#) below in this documentation.

Rater

■ Caches

A Rater is an instance dedicated to the available charging, recharging and refilling operations. Raters cannot be directly contacted by a client application and are thus considered as **private** instances. These instances are used to handle all the requests coming from the master Dispatcher.

According to the incoming requests, Raters can be considered as:

- **Stateful** Raters, when performing operations which alter the content of databases (statefull charging services)
- **Stateless** Raters, when performing operations which do not modify databases (stateless rating services)

The Raters are also responsible of managing the send and resend of spending status reports.

Managed caches

To ensure data consistency and improve the global performance of SAP CC, Raters handle multiple data structures using cache mechanisms. These cached structure are stored in memory and mostly concern the following business objects:

- Catalog data (offers, charge plans, charges, translation tables, tier tables, mapping tables, range tables, pricing macros, and so on)
- Provisioning data (contracts and associated allowances, subscriptions, subscriber accounts, subscriber mapping tables, subscriber range tables, and so on)
- Taxation data, for tax computation purposes
- Charging sessions and associated histories
- Shared allowances and distributed counters

**NOTES**

For further information about the cached structures, refer to the [adequate section](#) below in this documentation.

Updater

- [Active/backup](#)
- [Caches](#)

An Updater is a **public** instance dedicated to the provisioning operations used to manage customers' data. This instance can :

- Interact with client applications (such as the Core Tool user interface) using the XML over HTTP communication channel
- Communicate with the SAP CC CTS plugin using the SOAP over HTTP communication channel
- Communicate with the other instances through Dispatchers
- Communicate with the Updater instances from other SAP CC systems using the SOAP over HTTP communication channel in the context of the Catalog Transport feature

Active/backup updaters

Like Dispatchers, multiple Updaters can be configured in the static instance map. The difference with Dispatchers concerns the number of running instances: only one Updater can be started at a given time. This running Updater is then considered as the "active" Updater, and all the other non-running Updaters are then considered as "backup" Updaters. If the active Updater fails, client applications have to contact one of the backup Updaters.

**NOTES**

- All declared Updaters are pre-started in order to inform the master Dispatcher that they exist and can thus be used in case of failure of the active Updater
- Distinction between active and backup modes is made by the master Dispatcher. The active Updater corresponds to the first element which contacted the master Dispatcher. If the active Updater fails, the master Dispatcher chooses an element among the list of available backup Updaters and starts it
- Updaters do not provide any load balancing mechanism

Managed caches

To ensure data consistency and improve the global performance of SAP CC, Updaters handle multiple data structures using cache mechanisms. These cached structures are stored in memory and concern the following business objects:

- Catalog data, (offers, charge plans, charges, translation tables, tier tables, mapping tables, range tables, pricing macros, and so on)
- Provisioning data (contracts (without allowances), subscriptions (without counters), subscriber accounts, subscriber mapping tables, and subscriber range tables)
- Taxation data (to set up charge plans and provider contracts, offers and subscriptions)

**NOTE**

For further information about the cached structures, refer to the [adequate section](#) below in this documentation.

Taxer

■ Caches

A Taxer is a **private** instance dedicated to real-time U.S. Telco taxes calculation. As Taxers are only used to perform calculations, they do not impact any persistent data, and are thus considered as stateless instances. They encapsulate the taxation features provided by the BillSoft EZTax tax system and required by Raters for taxes computation purposes.



NOTES

- As Raters handle VAT taxes computations, Taxers are only required when U.S. Telco taxes are used. As a consequence, Taxers are considered as optional instances
- Taxers take a tax information as input and can output up to 15 tax amounts per transaction according to the applied taxation rules
- Taxers cannot be directly contacted by a client application. All taxation requests thus come from the Dispatcher instance
- To ensure better performances, multiple Taxers can be deployed to implement load balancing capabilities. As a consequence, communication with these deployed Taxers is then performed using a round-robin mode

Managed caches

As Taxers only represent an encapsulation of the BillSoft EZTax tax system, no specific data structures are cached.

Bulkloader

■ Caches

Charging operations performed by Raters lead to the production of charged item or refill record data files containing different information related to transactions. A Bulkloader is an asynchronous **private** instance used to:

- Load these charged item or refill record data files
- Convert the loaded charged items or refill records into billable items according to rules defined in the billable item mapping objects
- Send the converted data to SAP Convergent Invoicing for invoicing and billing purposes

To improve the global performance of these import operations, Bulkloaders are periodically triggered and use a bulk execution mode.



NOTES

- No redundancy mechanism has been implemented for Bulkloaders. To ensure performances and avoid latency, **it is recommended to deploy one Bulkloader per Rater**
- Bulkloaders are only able to manage charged item or refill record data files produced by Raters. They are only deployed when SAP CC is configured to work in conjunction with SAP Convergent Invoicing. For further information about bulk loading operations, refer to the description of the [Data Files Bulk Loading](#) feature in this documentation

Managed caches

To improve performance during bulk loading operations, Bulkloaders handle multiple data structures using cache mechanisms. These cached structures are stored in memory and concern the following business objects:

- Chargeable item classes, charged item classes and refill record classes, which contain the different properties of the defined chargeable items, charged items or refill records
- Consumption item mappings, which corresponds to mapping data between chargeable item classes and consumption item classes
- Billable item mappings, which corresponds to mapping data between charged item classes or refill record classes and billable item classes defined in SAP Convergent Invoicing

Synthesis

The following table summarizes the communication with and/or between the different instances:

Instance	XML over HTTP	Packets over TCP/IP	Messages over UDP
Dispatcher	■	■	■
Rater		□	
Guider		□	
Updater	■	□	
Taxer		□	
Bulkloader		□	

- Contacted by client applications
- Contacted by other server instances
- Contacted by client applications and other server instances



EXAMPLES

- Updaters can be contacted by client applications using the XML over HTTP communication channel, and by other instances using the Packets over TCP/IP communication channel
- Bulkloaders can only be contacted by other server instances using the Packets over TCP/IP communication channel

All these instances have been specifically designed to ensure the stability and performance of the global platform during pricing, charging and refilling operations. Furthermore, this architecture simplifies the scalability of SAP CC by giving the possibility to:

- Update deployed elements and/or upgrade them without any manual intervention
- Hot modify configuration parameters
- Add or remove instances without any downtime

The different instances of the Core Server are considered as:

- Public, when they can directly be contacted by an external system such as a client application
- Private, when they can only be contacted through another instance

All the information related to these instances is stored in a structure named "instance map". According to the type of concerned server instances, this instance map can be:

- A static structure, retrieved from the “INSTANCE_MAP” table of the Core database and containing pre-configured information related to Dispatchers and Updaters
- A dynamic structure, handled in memory by Dispatchers, and representing a version of the static structure enriched with information related to all started instances (Guiders, Raters, Taxers and Bulkloaders)

When SAP CC starts, the SAP startup framework performs the following operations:

- All the instances declared during the installation procedure are launched
- The starting instances analyze the `boot.config` file to get required information:
 - Core database connection information, used by Dispatchers
 - System identifier plus Dispatchers boot list or system identifier + UDP information (with possibly the Dispatchers list in case of discovery failure), used by Updaters
 - System identifier plus UDP information, used by Guiders, Raters, Taxers and Bulkloaders
- Dispatchers connect to the Core database in order to retrieve the static instance map and thus initialize their dynamic instance map which will be enriched by the master Dispatcher during the registration of the other instances
- Updaters, Guiders, Raters, Taxers and Bulkloaders try to connect to a Dispatcher using the discovery information:
 - If no Dispatcher can be discovered until a timeout occurs, the SAP startup framework restarts the concerned instances until a Dispatcher can be contacted
 - If a Dispatcher is discovered, the starting instances registers on it and becomes up and running



NOTE

The `boot.config` file contains the following parameters:

- `SQLHELPER_JDBC_URI`, which corresponds to the valid URL of the Core database
- `SQLHELPER_LOGIN`, which corresponds to the login used to connect to the Core database
- `SQLHELPER_PASSWORD`, which corresponds to the password used to connect to the Core database
- `SYSTEM_ID`, which corresponds to the identifier of the system where the instances must be deployed
- `DISCOVERY_MULTICAST_ADDRESS`, which corresponds to the IP address (IPv4 or IPv6) used for the UDP discovery mechanism
- `DISCOVERY_PORT`, which corresponds to the port used for discovery purposes

The static instance map contains the following properties, which are associated to a given instance identifier:

- `HttpSecure`, `HttpHost` and `HttpPort`, used for communications based on the XML over HTTP communication channel, and corresponding respectively to:
 - The secure mode
 - The hostname or IP address (IPv4 or IPv6) of the concerned instance
 - The port of the instance
- `WsSecure`, `WsHost` and `WsPort`, used for communications based on the SOAP over HTTP communication channel, and corresponding respectively to:
 - The secure mode
 - The hostname or IP address (IPv4 or IPv6) of the instance
 - The port of the instance
- `ExternalSecure`, `ExternalHost` and `ExternalPort`, **used by external client applications** for communications based on the Packets over TCP/IP communication channel, and corresponding respectively to:
 - The secure mode
 - The hostname or IP address (IPv4 or IPv6) of the Dispatcher

- The port of the Dispatcher
- ExternalSecure, ExternalHost and ExternalPort, **used by the other instances** for internal communications based on the Packets over TCP/IP communication channel, and corresponding respectively to:
 - The secure mode
 - The hostname or IP address (IPv4 or IPv6) of the Dispatcher
 - The port of the Dispatcher

The following table shows the availability of these properties according to the specified instance:

Property	Dispatcher	Updater
HttpSecure	■	■
HttpHost	■	■
HttpPort	■	■
WsSecure	■	■
WsHost	■	■
WsPort	■	■
ExternalSecure	■	
ExternalHost	■	
ExternalPort	■	
InternalSecure	■	
InternalHost	■	
InternalPort	■	

The content of this static instance map is loaded into the Core database. As a consequence, the Core database must be updated when the static instance map is modified.

BART Server

The BART Server component is an injector dedicated to **batch** usage charging operations. This component is in charge of the following processes:

- Chargeable Items Acquisition, which defines the way to load incoming CDRs into a dedicated database. According to the defined policies, duplicate CDRs can be detected and managed (e.g. rejected, marked as corrected later, and so on)
- Chargeable Items Consolidation, which is used to retrieve additional information from the Core database in order to enrich the incoming CDRs
- Batch execution of the Chargeable Items Charging process, which consists in charging a set of CDRs, subscription per subscription
- Chargeable Items Rerating, which gives the possibility to charge again a set of CDRs when errors such as human error occurred in the rating engine (e.g. customers who subscribed to an erroneous charge plan or offer)

The BART Server is delivered with a set of desktop and console applications which give the possibility to:

- Administer its behavior
- Manage the handled master data

The following table provides some detailed information related to these applications:

BART Tool BART Tool is a desktop application dedicated to the BART Server supervision. Written entirely in Java and respecting the look & feel of the Core Tool user interface, this application provides graphical interfaces used to monitor the BART Server and ease some operations such as:

- Acquisition and charging sessions management
- CDRs and CDR groups management
- Jobs administration

BART Setup Tool Like the Core Server, a BART Setup Tool is also available for the BART Server. This command-line tool has been introduced in the SAP CC 5.0 release, and only gives the possibility to manage the security settings related to communication channels securing.



NOTES

For further information about the BART Setup Tool and associated commands, refer to the [SAP Convergent Charging Configuration Guide](#) and [SAP Convergent Charging User Interfaces](#) documentations.

BART+ BART+ is a console application dedicated to the management of the BART Server. It works the same way as Admin+ and performs operations such as:

- Getting or setting the values of some parameters used by the BART Server
- Updating the level of log messages generated by the BART Server
- Monitoring acquisition sessions
- Intercepting messages and redirecting them to a specified file
- Pinging the server

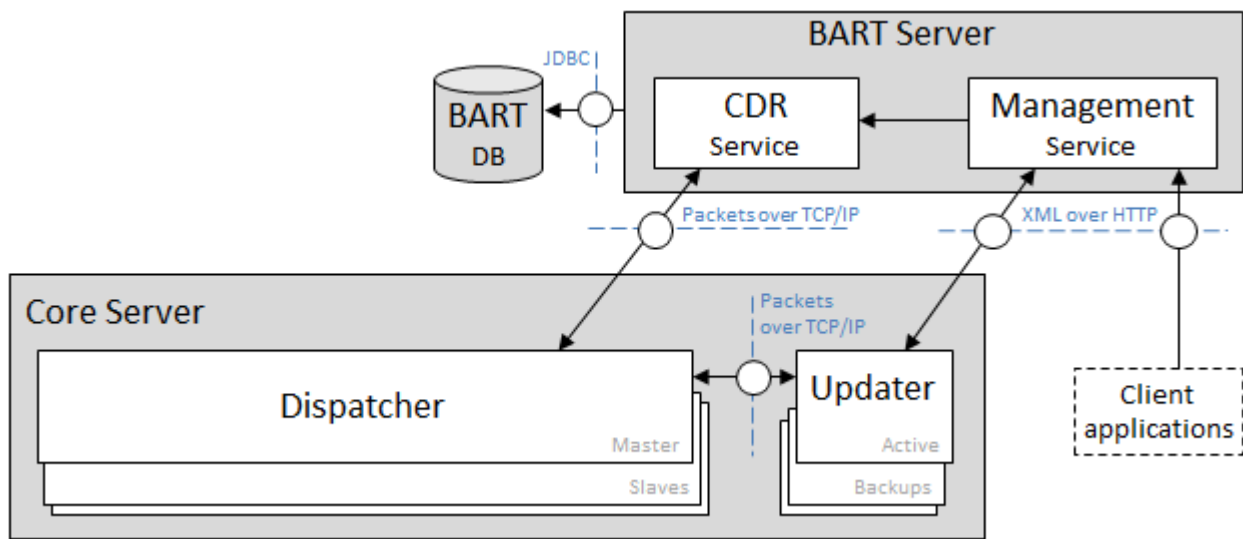


NOTE

For further information about BART+ and its associated commands, refer to the [SAP Convergent Charging Configuration Guide](#) and [SAP Convergent Charging User Interfaces](#) documentations.

The architecture of the BART Server is based on the standard 3-tier architecture and relies on 2 different services:

- A CDR service, used to:
 - Acquire CDRs from client applications or mediation systems (such as the IEC)
 - Perform operations such as consolidation, duplicate detection, and so on
 - Communicate with the Core Server to perform batch charging operations
- A management service, used to handle XML messages sent by client applications (including BART Tool) to perform operations on charging sessions (such as session start, stop, and so on) or execute configuration and administration operations



NOTE

The BART Server does not behave as a mediation module, but can be configured to work in conjunction with the IEC in order to acquire CDRs coming from various sources.

The communication inside and/or with the BART Server relies on 2 different communication channels:

- XML over HTTP, used to transport proprietary XML messages (named “HCI envelopes”)
- Packets over TCP/IP, used to transport proprietary TCP packets

These protocols and their associated transported data are similar to those used by the Core Server. For further information about these protocols, refer above.

Diameter Server

The Diameter Server component represents a message translator which is used to convert credit-control messages coming from a Diameter client application into charging messages that the Core Server is able to handle. It behaves as an online mediation system:

- Providing credit-control capabilities to:
 - Control services and manage prepaid balances in real-time
 - Notify customers when balance thresholds are reached
 - Terminate a service session when a balance is depleted
 - CCA enrichment with attributes specific to a given network element (VSA)
 - Diameter security mechanisms like TLS or IPSec
 - Multiple services credit-control in a single session
- Unable to handle the following features:
 - Tariff-Time-Change mechanism, used to charge a service differently according to time
 - Dynamic peer discovery mechanism
 - Re-authentication or re-authorization requests, used to check whether the user is still engaging the service, and if not, to remove the pending session to avoid further charging command codes
 - Session termination requests initiated by the Diameter Server (such as Abort-Session-Request), used to close a pending session

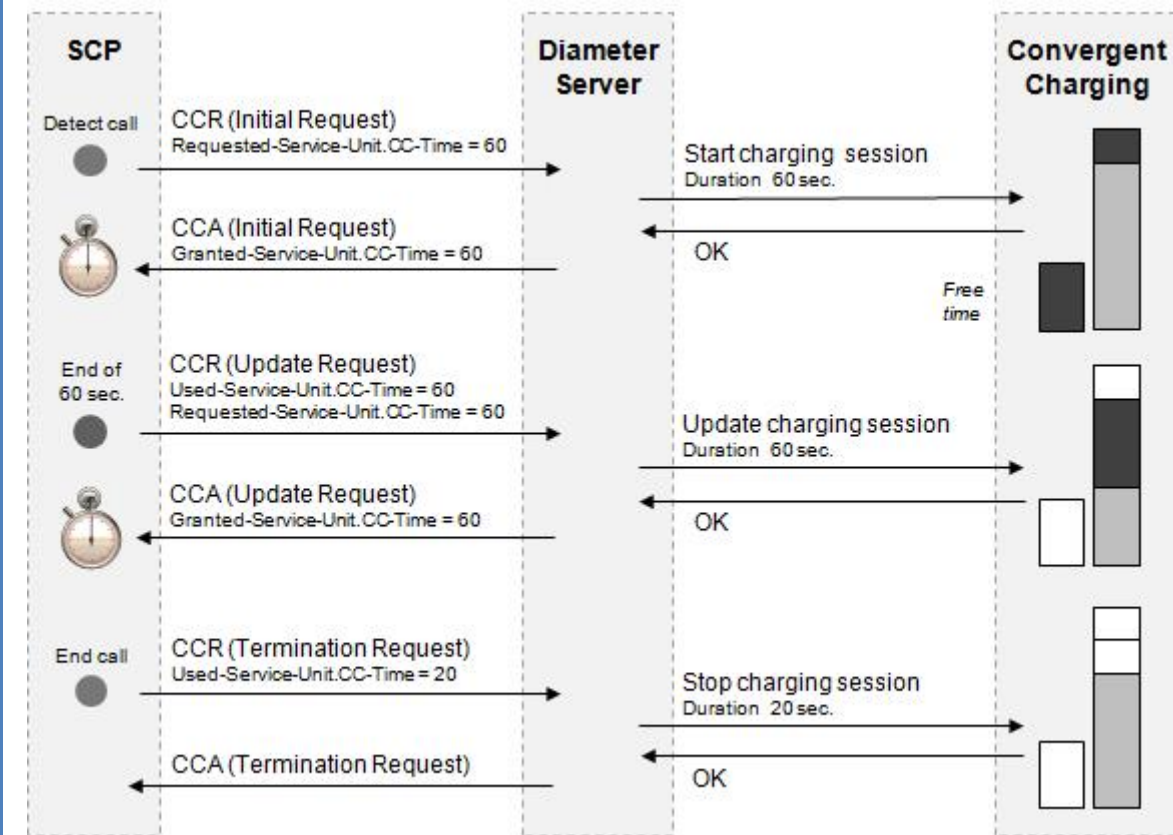
- Relay, Proxy, Redirect Translation of Diameter messages

The Diameter Server respects the following specifications:

- RFC 3588, which describes the Diameter base protocol
- RFC 4006, which describes Diameter credit-control applications
- 3GPP TS 32.299, which describes Diameter online charging applications (Ro interfaces)

EXAMPLE

The following schema describes a SCUR call flow between a SCP, the Diameter Server component, and SAP CC:



MESSAGE FLOW USE CASES

Use case 1: Voice call with re-authorization and low credit

Description	SCP	Diameter Server
Call detected	CCR (Initial Request) Requested-Service-Unit.CC-Time = 60	→
Enough credit to allocate 60s		← CCA (Initial Request) Granted-Service-Unit.CC-Time = 60
End of quota. Re-authorization request	CCR (Update Request) Used-Service-Unit.CC-Time = 60 Requested-Service-Unit.CC-Time = 60	→
Enough credit to allocate 15s		← CCA (Update Request) Granted-Service-Unit.CC-Time = 15 Final-Unit-Action = TERMINATE

End of last quota.	CCR (Termination Request) Used-Service-Unit.CC-Time = 15	→	
Call ended		←	CCA (Termination Request)

Use case 2: Voice call with no credit

Description	SCP		Diameter Server
Call detected	CCR (Initial Request) Requested-Service-Unit.CC-Time = 60	→	
Not enough credit. Call rejected		←	CCA (Initial Request) Result-Code = DIAMETER_CREDIT_LIMIT_REACHED

Use case 3: Non-connected voice call

Description	SCP		Diameter Server
Call detected	CCR (Initial Request) Requested-Service-Unit.CC-Time = 60	→	
Enough credit to allocate 60s		←	CCA (Initial Request) Granted-Service-Unit.CC-Time = 60
Non connected	CCR (Termination Request) Used-Service-Unit.CC-Time = 0	→	
Reservation cancelled		←	CCA (Termination Request)

Use case 4: Disconnected voice call

Description	SCP		Diameter Server
Call detected	CCR (Initial Request) Requested-Service-Unit.CC-Time = 60	→	
Enough credit to allocate 60s		←	CCA (Initial Request) Granted-Service-Unit.CC-Time = 60
<i>...Call disconnected...</i>			
No call session available	CCR (Termination Request) Termination-Cause = DIAMETER_SERVICE_NOT_PROVIDED	→	
Reservation cancelled		←	CCA (Termination Request)

Use case 5: SMS without reservation

Description	SCP		Diameter Server
SMS sending	CCR (Event Request, Direct Debiting) Requested-Service-Unit.CC-Time = 60	→	
Enough credit to send the SMS. SMS delivered		←	CCA (Event Request) Granted-Service-Unit.CC-Service-Specific-Units=1

Use case 6: SMS with reservation

Description	SCP		Diameter Server
SMS sending	CCR (Initial Request) Requested-Service-Unit.CC-Service-Specific-Units=1	→	
Enough credit		←	CCA (Initial Request) Granted-Service-Unit.CC-Service-Specific-Units=1
SMS delivery	CCR (Termination Request)	→	

	Used-Service-Unit. CC-Service-Specific-Units=0		
SMS delivered		←	CCA (Termination Request)

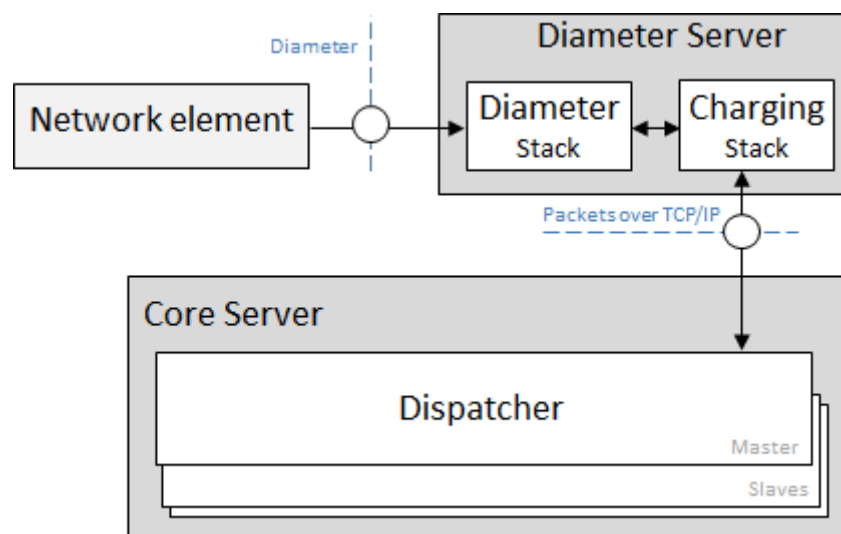
The architecture of the Diameter Server is built upon the following set of elements:

- **Diameter Stack**, which is provided by the Traffix Systems company and implements the Diameter base protocol (RFC 3588/3588bis standards, part of the RFC 4006 dedicated to Diameter Credit Control Applications)
- **Charging Stack**, which handles connections with the Core Server in order to send charging requests and receive the related answers
- **Service Dictionary**, which gives the possibility to store all the descriptions (including the AVP mapping) of each managed service
- **AVP Dictionary**, which gives the possibility to store the AVP descriptions related to specific vendor network elements
- **CCR Handler**, which:
 - Receives CCRs coming from the SCP through the Diameter Stack
 - Translates these CCRs to charging requests using the AVP mapping described in the Service Dictionary
 - Sends these charging requests to the Core Server through the Charging Stack
- **Charging Answers Handler**, which:
 - Receives charging answers coming from the Core Server
 - Translates these charging answers them to CCA
 - Sends these CCS to the SCP through the Diameter Stack

The communication inside and/or with the Diameter Server relies on 2 different communication channels:

- Diameter, used for credit-control purposes
- Packets over TCP/IP, used to transport proprietary packets and communicate with the Core Server

For further information about the Packets over TCP/IP communication channel, refer to the [SAP Convergent Charging Security Guide](#) documentation. For further information about the Diameter protocol, consult the RFC 3588 and RFC 4006 specifications.



IEC

The IEC is an optional component used to connect SAP CC to external systems such as back office, legacy applications, and so on.

This component is delivered with a desktop application named “CAT Tool” which gives the possibility to interact with the Core Server and access to its usage data records. The different data transfers are described into scenarios which represent a sequence of basic actions such as:

- Importing/exporting data from/to a file or data container
- Modifying data
- Executing a given Java class

This desktop application provides a set of components which give the possibility to fulfill the different actions which are necessary to perform data transfers.



NOTE

For further information about CAT Tool, refer to its dedicated [SAP Convergent Charging Online Help](#) documentation.

Databases

SAP Convergent Charging deals with different databases used by the different deployed components. All these databases contain data handled by specific processes and functions whose performances require configuration and tuning.

Core Database

The Core database contains the different data related to the business objects handled by the Core Server, such as price plan elements, subscriptions, offers, accesses, and so on. To ensure performances, the rating engine needs to perform a huge number of connections and transactions (with low latency) to this database. As a consequence, it is highly recommended to configure the Core database as an On-Line Transaction Processing (OLTP) database in order to execute real-time processes.

Session Database

The Session database contains the different data related to the business objects handled by SAP CC Core Server during session-based charging operations.

Cockpit Database

The Cockpit Database is an embedded persistent database that is used by Cockpit to:

- Store user settings (launchpad configuration, apps variants, and so on)
- Store drafts for objects created in the different apps (mapping table classes, mapping tables, and so on)
- Act as a SQL cached structure that decreases the solicitation of the Core database, by storing temporary data used throughout the apps



NOTE

As the Cockpit database is an embedded database, no specific installation is required. For further information, refer to the “Configuring Cockpit” section of the [SAP Convergent Charging Tuning Guide](#) documentation.

BART Database

The BART database contains the different data related to the business objects handled by the BART Server, such as CDRs, CDRs status, charging sessions, and so on.

IEC Database

The CAT Tool provides a component named “Export to Database Action” which gives the possibility to export the created chargeable items to the IEC database for futures actions (using a complementary import component).

Database technologies

SAP Convergent Charging supports the following database technology:

- SAP HANA
- SAP ASE (aka Sybase ASE)
- Oracle
- Microsoft SQL Server
- IBM DB2

Refer to the PAM information for more detailed information about the supported versions of RDBMS.

Database editions

The default edition of the Oracle and Microsoft SQL Server RDBMS is named “Standard Edition”, and is suited for small and medium sized businesses. According to the needs, this edition may be inadequate because of limitations related to hardware platforms, such as:

- Limited number of CPUs
- Unavailable clustering features
- Memories size (influencing simultaneous connections capabilities)
- Hard disks architecture and performances



CAUTION

The standard edition of the Microsoft SQL Server RDBMS is not supported by SAP CC 5.0.

Another edition named “Enterprise Edition” represents the advanced edition of Oracle and Microsoft SQL Server RDBMS, suited for larger businesses looking for advanced database features such as:

- Replication for distributed systems
- Failover mechanisms to ensure robustness
- Advanced queuing mechanisms
- Transportable tablespaces management (Oracle only)
- Improved querying mechanisms
- Parallelized operations such as index management, backup/recovery
- Data partitioning to improve performances
- Advanced mirroring features improving HA mechanism

Database partitioning

All databases handled by SAP CC require specific configuration according to the business operations which need to be performed on contained data. To ensure performances and stability towards concurrent accesses, databases data

have been partitioned in order to isolate them during operations executions. Two partitioning methods have been implemented according to the performances needs:

- **Business partitioning**, based on the association of table records and partition identifiers. This method ensures that server instances associated to a given group of partitions can only execute operations concerning data associated to these partitions
- **Storage partitioning** (made up with sub partitions), based on data regrouping according to criteria such as recording date. This method ensures performances on operations dealing with tables containing huge quantities of business objects



NOTES

- SAP HANA databases do not support any partitioning mechanism.
- IBM DB2 databases only support the business partitioning concept.

Partitioning availability

The following table shows the availability of the partitioning methods according to the RDBMS edition:

RDBMS	Business partitioning		Storage partitioning	
	Core database Session database	BART database	Core database Session database	BART database
SAP ASE				
Enterprise edition (with partitioning license)	■			■
Oracle				
Standard edition				
Enterprise edition	■			■
MS SQL Server				
Enterprise edition	■			■
IBM DB2				
Enterprise edition	■			

Business partitioning

Business partitioning only concerns the Core database, and consists in partitioning data according to a business partition identifier. This kind of partitioning is used to group and isolate data to handle concurrent accesses and guarantee performances (for example during search operations which operate within large data scopes).

The Business Partitioning implemented for Oracle databases is based on a `PARTITION_ID` column of some Core database's tables. This partition identifier belongs to the following ranges:

-]0, 480[for data managed by Raters, using 12 groups of 40 possible values
-]0,240[for data managed by Guiders, using 12 groups of 20 possible values

This partition identifier is implemented into various database tables according to the following pattern:

```
CREATE TABLE XXX (
...
PARTITION_ID DECIMAL(4) NOT NULL,
) PARTITION BY RANGE(PARTITION_ID) (
PARTITION ACM_P ONE VALUES LESS THAN (20),
```

```

PARTITION ACM_P_TWO VALUES LESS THAN (40),
...
PARTITION ACM_P_ELEVEN VALUES LESS THAN (220),
PARTITION ACM_P_TWELVE VALUES LESS THAN (240)
);

```

In most cases, the business partitioning is combined with a local indexation (each index is local to the partition). All the tables' related indexes are declared as `LOCAL INDEX`.

The Business Partitioning implemented for MS SQL Server databases is similar to the Oracle implementation. The difference with the Oracle implementation concerns the partition identifier which only belongs to the]0, 480[range using 12 groups of 40 possible values.

Storage partitioning

Storage partitioning concerns the `CDR` table of the BART database, because of the huge volumes of stored data. This kind of partitioning has been designed to improve performances of operations which handle high quantities of data, for example purge and/or archive operations (a `DROP PARTITION` command is far more efficient than a bulk `DELETE` command based on a date range) which deal with no more used data.

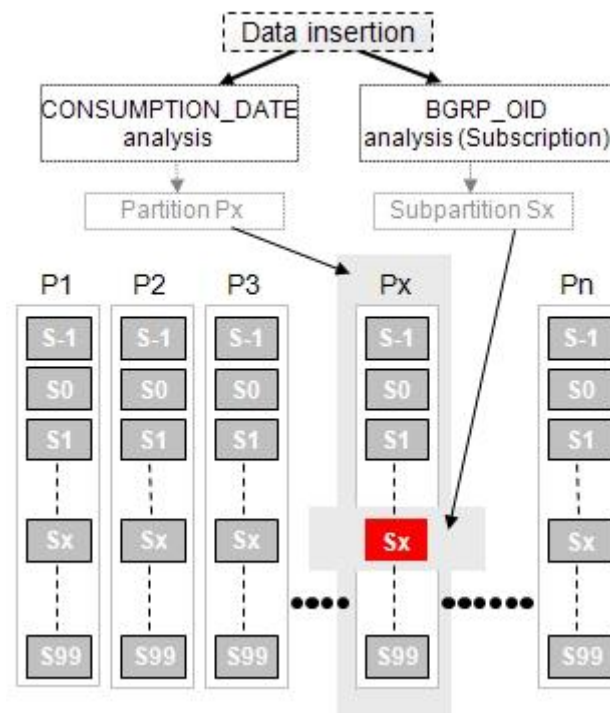
The Storage partitioning implemented for Oracle databases (also called "Composite partitioning") uses the **Range** partitioning method, in conjunction with "subpartitions" created using a **List** partitioning method. All these managed partitions are stored in the same tablespace/filegroup and every tablespace/filegroup is independent.



NOTES

- The **Range** partitioning method is based on temporal intervals whereas the **List** partitioning method is based on a finite list of values
- The default **Range** partitioning method is based on daily intervals, with a default 40 days configuration extendable up to 70 days at installation time
- The default **List** partitioning method splits these 40 partitions into 100 subpartitions, which gives the possibility to manage a total of maximum **100 x 70 = 7000** partitions
- If the Chargeable Items Recharging feature is enabled, the configuration of the CDRs keeping days must be equal to the retention period used for recharging operations
- For further information about the sizing rules which are available for the BART database, refer to the [SAP Convergent Charging](#) documentation

The **Range** partitioning concerns the `CONSUMPTION_DATE` field and the **List** partitioning concerns the `BGRP_OID` field which is the batch group identifier allocated at subscription creation time and belongs to the range [-1, 99] (-1 meaning "no batch rating group defined"):

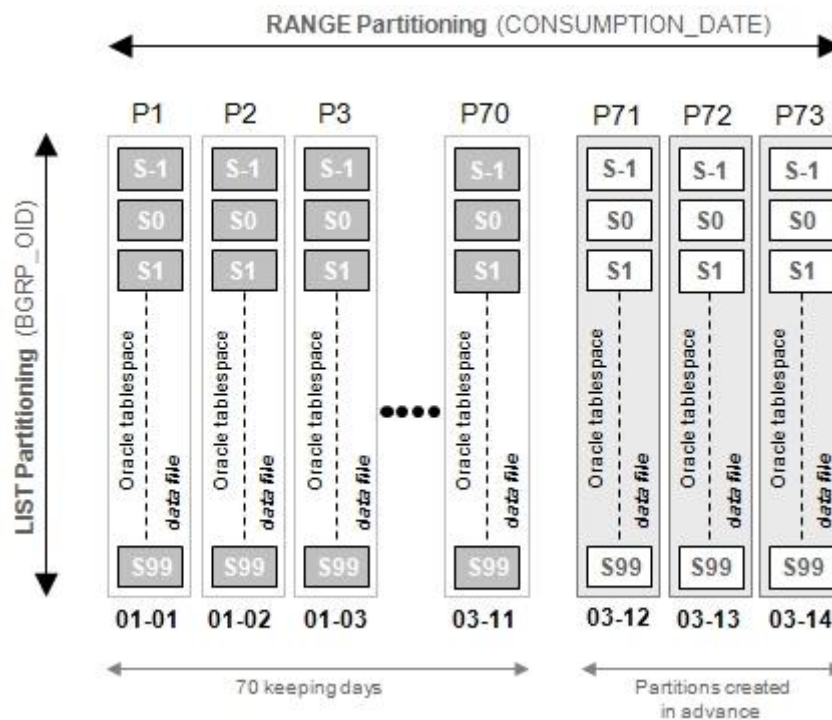


List partitioning

 **NOTE**

All subpartitions S-1...S99 of a given partition are located into one or more Oracle tablespaces (which represent logical storage elements). Every tablespace is physically recorded into a data file stored inside a directory of the file system.

The following schema gives an idea on the database storage associated to this Storage partitioning. This schema represents a snapshot of the database storage at the 11th of March 2007 with 70 transactions keeping days:



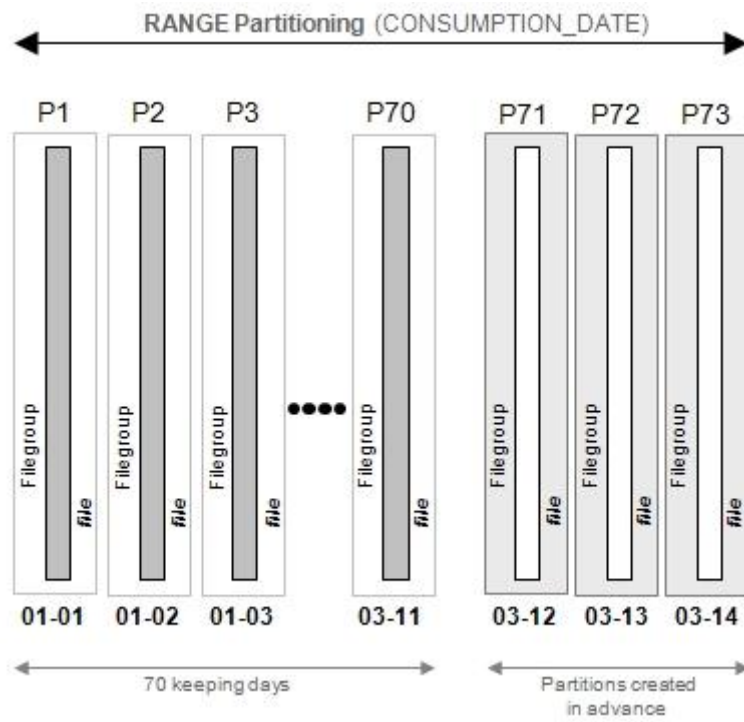
Range and List partitioning

The Storage partitioning implemented under Microsoft SQL Server database is slightly different from the Oracle implementation, because **the maximum number of managed partitions is limited to 1000**. As a consequence, Microsoft SQL Server only implements the **Range** Partitioning for the CDR table of the BART database.

NOTES

- The **Range** partitioning method is based on temporal intervals
- The default **Range** partitioning method is based on daily intervals, with a default 40 days configuration extendable up to 70 days at installation time. This configuration thus gives the possibility to manage a maximum total of **70** partitions
- If the Chargeable Items Recharging feature is enabled, the configuration of the CDRs keeping days must be equal to the retention period for rating
- For further information about the sizing rules which are available for the BART database, refer to the [SAP Convergent Charging](#) documentation

The following schema describes the partitioning implemented on Microsoft SQL Server databases:



Range partitioning

6 Master Data



















SAP Convergent Charging deals with different sets of master data related to:

- Service providers, which consist in charging and refilling data structures required for building the business logic used by the rating engine
- Customers, which consist in data structures required to apply the service provider's business logic to the account(s) of a customer
- Input and output transactional data, which represent the data structures that users or external systems can deal with

The availability of these master data depends on the implementation of SAP CC, which specifies a working set of objects that must be used when configuring a pricing catalog:

- **Provider Contracts only:** This model has been specifically designed for SAP CC 3.0 and 4.x releases, to fit an integrated scenario with the SAP ERP/FI-CA component of the SAP Business Suite. This model ensures that all the used master data will be compatible with the SAP ERP systems
- **Provider contracts and Subscriptions:** This model has been designed for the implementations of SAP CC which have been migrated from the 2.0 to the 3.0 release. It behaves like the "Provider Contracts only" model, but also gives the possibility to continue using old master data such as subscriptions, charges (in their previous versions), and offers

The following table lists these master data and shows their availability according to the configured model:

Master data	Provider Contracts and Subscriptions	Provider Contracts only
Core Server		
Master Data for a Service Provider		
 Product	■	■
 Service Provider	■	■
 Catalog (Pricing)	■	■
 Chargeable item	■	■
 Charged item	■	■
 Billable item	■	■
 Consumption item	■	■
 Charge ¹	■	■
 Counter	■	■
 Price plan	■	■
 Charging plan	■	■
 Offer	■	
 Aggregation Policy	■	■
 Charge plan	■	■
 Customized charge	■	■
 Refill item	■	■
 Refill record	■	■
 Refill logic	■	■

↗ Refill plan	■	■
↗ Customized refill logic	■	■
↗ Monitoring plan		■
↗ Allowance Event		■
↗ Allowance Logic		■
↗ Customized Allowance Logic		■
↗ Allowance Plan		■
↗ Allowance Interface		■
↗ Allowance Definition Table and Allowance Definition Table Class		■
↗ Change List	■	■
<i>Pricing elements for the decision trees</i>		
↗ Mapping Table Class and Mapping Table	■	■
↗ Price Table	■	■
↗ Range Table Class and Range Table	■	■
↗ Tier Table	■	■
↗ Translation Table	■	■
↗ Pricing Macro	■	■
Master Data for the End Customers		
↗ Subscriber account	■	■
↗ Subscriber mapping table		■
↗ Subscriber range table		■
↗ Subscription	■	
↗ Provider contract	■	■
↗ Contract item	■	■
↗ Access	■	■
↗ Allowance		■
BART Server		
↗ CDR (Consumption Detail Record)	■	■
Diameter Server		
↗ CCR (Credit Control Request)	■	■
↗ CCA (Credit Control Answer)	■	■

¹ The content of the charges varies according to the model. SAP CC 3.0 charges are not exactly the same as the SAP CC 2.0 ones

Master Data for the Service Provider

Product

Modeled within a CRM application or another provisioning system, a product represents a set of marketable services that end customers can purchase and use. Every product is made up with a set of master data which:

- Describes and structures a service
- Is transmitted to the rating engine for charging purposes

This master data are extended by SAP CC (in terms of pricing information) in order to dynamically compute price amounts related to the usage of a service. This information is stored in a pricing catalog owned by the service provider.

Service Provider

A service provider is a legal entity that offers commercial products (such as mobile phone, telephone call and Internet access) to business partners. It represents an actor who has long term business agreements with these business partners.

In SAP Convergent Charging, a service provider is set up as a set of pricing elements which enable charging customer services. Master data for a single service provider can be configured with several marketable services. For each marketable service, it is necessary to:

- Set up master data such as chargeable item classes and charges
- Combine these master data in charge plans or in offers

The business partners are represented in SAP CC by a subscriber account and by a provider contract or a subscription.



NOTE

The configuration of the master data for the service provider highly depends on the integration of SAP CC with:

- A mediation system, which sends to SAP CC the different usage data which must be charged
- A billing system, which will receive the charged items generated by SAP CC in order to produce invoices in a financial system

Very good skills about the implementation of these two interfaces are prerequisites for the pricing operations of marketable services in SAP CC.

Catalog

A catalog represents a container object which is used to group all the pricing elements related to one or several marketable services of a service provider. For organization and security reasons:

- An SAP user can be restricted to a given catalog
- Several catalogs can be used to restrict the configurations of master data related to a subset of services



NOTES

- When SAP CC is integrated with the SAP ERP/FI-CA component of the SAP Business Suite, only one pricing catalog for a service provider can be configured when managing recurrent fees with price tables
- Users' visibility and management of master data depends whether they are associated to a given catalog or not: an SAP user which is associated to a catalog can only access to the master data contained in this catalog

Chargeable Item

A chargeable item is the technical result of an external action or usage (such as a mobile phone, telephone call or Internet service) that the consumer carries out in the framework of marketable services. Massively sent to SAP CC by

external mediation systems for charging purposes, this technical transactional data record contains the properties and their associated values required by the different charging operations.

Chargeable item class

A chargeable item class contains a list of properties which characterize each usage event used inside price plans and/or charging plans for charging purposes. These properties can be:

- Default properties, used to provide basic information about the consumption of the service, and whose definition cannot be modified
- User-defined properties, created and modified by users according to their needs

A chargeable item class thus represents the description of the chargeable items which have the same structure.

Chargeable item package

A chargeable item package is a set of chargeable item classes, which gives the possibility to apply a charge on multiple chargeable items at the same time.

Charged Item

A charged item is the technical result of a chargeable item which has been charged. It contains information such as:

- The charging date
- The identifier of the charged account, which references an account managed internally into SAP CC or externally into a third-party billing system such as the SAP ERP/FI-CA component of the SAP Business Suite
- The charged amount, with or without generated tax data (according to the performing of the tax calculation process)

Charged items are generated by the rating engine at the end of the [Chargeable Items Charging](#) and the [Prepaid Accounts Refilling](#) processes. The generation of charged items is performed according to specific configurations set up in the charges or allowance logic objects inserted into an offer, a charge plan, or an allowance plan.

Charged item class

A charged item class represents a list of field descriptions which gives the possibility to transform a charged transaction into a charged item and thus configure the generation of charged items sent to a third-party billing system. These field descriptions can be created from scratch or from predefined classes, and can be redefined at the offer, charge plan or allowance plan levels in order to fit specific needs (through charged item mapping described below).

Charged item mapping

When configuring offers, charge plans or allowance plans, users can redefine the values of the different properties of the associated charged item class. This redefinition is done through the charged item mapping part of offers, charge plans or allowance plans, by specifying a value or choosing another property of the charging context.

Billable item

A billable item represents a data record received by the SAP ERP/FI-CA component of the SAP Business Suite for billing and invoicing purposes. SAP CC can convert its generated transactional data (charged items and refill records) into billable items and send these records to SAP Convergent Invoicing.

Billable item class

A billable item class corresponds to the same concept as charged item classes (or refill record classes) but only refer to SAP Convergent Invoicing. They only describe the different properties of a billable item.

Billable item mapping

To ensure that the generated charged items will be correctly recorded into SAP Convergent Invoicing, the charged item classes (or refill record classes) can be mapped onto billable item classes. This mapping gives the possibility to transform a property into another one which can have a different name and/or a different format expected by in the targeted system.

Consumption item

A consumption item represents the converted version of a chargeable item, which is sent to SAP Convergent Invoicing for deferred charging and recharging purposes. The creation of consumption items relies on the same concepts and architecture than the creation of billable items, and the transfer from SAP CC to SAP CI is also performed asynchronously by Bulkloaders.

Consumption item class

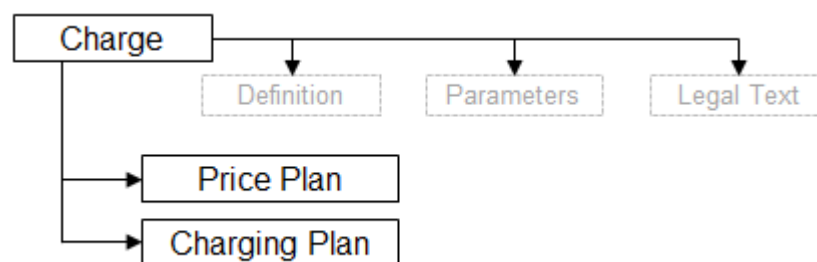
Consumption item classes are similar to billable item classes and are also defined in SAP Convergent Invoicing. SAP CC charges the different properties of a mediation system which are configured in the consumption item classes of SAP CI. In SAP CC, each consumption item class is replicated as a chargeable item class. For more information on consumption item classes, refer to the documentation of SAP Convergent Invoicing.

Consumption item mapping

The chargeable item classes are mapped onto consumption item classes to ensure that the chargeable items generated by SAP CC will be correctly recorded into SAP CI. This mapping may thus transform a property into another one which can have a different name or a different format expected by the targeted system.

Charge

A charge represents a cost and a charging pattern linked to one or several customer services. The cost is described in a calculation algorithm and configured in the price plan within the charge, while the charging pattern is described in the charging plan.



Relationships between charges, chargeable item classes, charging plans and price plans are the following:

- Charges always contain a unique price plan and a unique charging plan
- A price plan describes cost calculations or other customer service related costs to be charged. These cost calculation may use properties coming from incoming chargeable items whose chargeable item classes may differ

- A charging plan describes the charging patterns which relate to the cost defined in the price plan. These charging patterns may use properties coming from chargeable items whose chargeable item classes may differ
- Several chronological versions of a price plan or a charging plan can coexist in the rating context (each version taking into account changes in product costs)
- To increase offer flexibility, a charge can be applied on different chargeable item classes

Master and dependent charges

Every charge is associated to a type which is determined during the charge creation and cannot be modified later. Every charge can be:

- A **master** charge: This kind of charge is not synchronized with other charges. It is independent of other charges combined in a charge plan or in an offer and thus has its own trigger and calculation modes. It represents neither an associated cost (such as commission, sponsorship) nor a derived cost (such as a discount on a basic price).
- A **dependent** charge: This kind of charge is synchronized with another charge and uses its own trigger and/or calculation modes. A dependent charge may depend on another dependent charge. These links are then translated into a series of descending dependencies. Dependent charges are arranged according to their priority in the charge plan or in the offer. They are also assigned to a role (commission, third-party billed commission, sponsorship, discount) depending on the business requirements to model.

NOTE

In an integrated *SAP Solution* scenario with the SAP CRM and the SAP ERP/FI-CA components of SAP Business Suite, if you need to manage payments to a business partner, SAP SE or an SAP affiliate company recommends that you only configure appropriate master charges in SAP CC with the *Revenue Sharing and Partner Settlement* function enabled into the SAP ERP/FI-CA component.

Refer to the [O2C Integration Guide](#) documentation for more information about partner settlement, customer billable item classes, and partner billable item classes.

Combination of charges

A charge is a reusable object which must be combined and customized in another business objects:

- In a charge plan for charging operations based on provider contracts
- In an offer for charging operations based on subscriptions

NOTES

- The 5.0 release of SAP CC provides two variants of the “Convergent Charging” business scenario. According to your integration scenario, you will use its global charging services based on:
 - Provider contracts (typical integration with an external CRM system)
 - Subscriptions (integration with various and heterogeneous external provisioning systems)
- In an integrated *SAP Solution* scenario with the SAP CRM and the SAP ERP/FI-CA components of SAP Business Suite, the convergent charging services are based on provider contracts

Counter

A counter is a variable mostly declared and defined in a charge, in a refill logic or in an allowance logic. Each counter has a name and an initial value. A counter is used to introduce a numerical property when modeling a decision tree (price plan, pricing macro, charging plan or refill logic).

All the necessary counters are finally created and stored in the business agreements (provider contract, subscription) between the service provider and its customers. A counter is used every time SAP CC performs charging and refilling operations at runtime and executes the calculation algorithms configured in the decision trees.

**NOTE**

An unlimited number of counters can be theoretically declared, but a huge number of counters can lead to decreasing performances.

Persistent and transient counters

Persistent counters are created, initialized and stored in the business agreement (provider contract, subscription). They are modified during charging and refilling operations. SAP CC permanently stores these values in database.

Transient counters are created during charging and refilling operations. They are only **temporary** stored in the memory of the system until the end of the calculation process.

**NOTE**

A transient counter can only be declared in a price plan of a charge or in an allowance logic.

Counter redefinition

The initial value of a counter can be redefined during the modeling of:

- A charge plan
- A refill plan
- An allowance plan
- An offer

This redefinition is also available at the creation of a subscription for a customer during the [Order Management](#) process.

**NOTE**

It is not possible to redefine the initial value of a counter when creating a provider contract. The external CRM or provisioning system is not supposed to redefine the initial values of such visible counters. By default, shared counters are initialized with a zero value.

Counter sharing

SAP CC gives the possibility to share the values of different counters created inside the business agreements. Shared counters must be specifically declared at the charge plan, refill plan or offer levels:

Sharing Level	Description	Configurations of Master Data for Service Provider
Subscriber Account	A counter can be shared between multiple provider contracts belonging to the same subscriber account	Configurations are necessary: A parent contract shares a counter with linked provider contracts.
Provider contract	A counter can be shared between multiples charges and refill logic included in different contract items	Configurations are necessary: <ul style="list-style-type: none"> ■ In the charge plan: a counter must be declared and set as visible by external systems

		<ul style="list-style-type: none"> ■ In the charges customized in this charge plan: a counter of each charge must be linked to the counter previously declared in the charge plan ■ In the refill plan: a counter must be declared ■ In the refill logic customized in this refill plan: a counter must be linked to the counter previously declared in the refill plan <p>The final configuration is necessary in the CRM or external provisioning system (see section about the provider contract for details).</p>
Contract item	A counter is shared between all the charges activated in a contract item referencing a charge plan	<p>Configurations are necessary:</p> <ul style="list-style-type: none"> ■ In the charge plan: a counter must be declared ■ In the charges customized in this charge plan: a counter of each charge must be linked to the counter previously declared in the charge plan
Subscription	A counter is shared between all the charges activated in the subscription	<p>Configurations are necessary:</p> <ul style="list-style-type: none"> ■ In the offer: a counter must be declared ■ In the charges customized in this offer: a counter of each charge must be linked to the counter previously declared in the offer
Sub-subscription	A counter is shared between all the charges activated in the sub-subscription	<p>Configurations are necessary:</p> <ul style="list-style-type: none"> ■ In the suboffer: a counter must be declared ■ In the charges customized in this suboffer: a counter of each charge must be linked to the counter previously declared in the suboffer

Counter creation in a provider contract

Persistent counters are not created during the creation of the provider contract. Each counter is automatically created when it is needed by SAP CC to perform a business process (charging, refilling or activation):

- For a usage charging: the system creates automatically and only the counters which are:
 - Declared in a charge
 - Needed during the execution of a usage rate included in the price plan
- For a usage refill: the system creates automatically and only the counters which are:
 - Declared in a refill logic
 - Needed during the execution of a usage refill included in this refill logic
- For the external or internal triggering of the activation process: the system creates the counters which are needed for the management of recurring rates or refills.

NOTE

A provider contract may miss some counters because they have not been required by the system.

Counter creation in a subscription

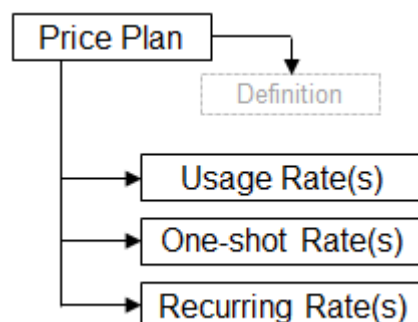
Persistent counters must be created during the creation of a subscription. They must be created at the appropriate location in the structured subscription (root, sub-element). Each element of the subscription which is supposed to use this shared counter must refer to it.

NOTES

- Counters values can be initialized, set, defined and redefined at different levels:
 - In the charge
 - In the charge plan, in the refill plan or in the offer
 - In the provider contract or in the subscription
- In a charge plan, the visibility of counters can be restricted to SAP CC when a counter has no meaning in the external CRM or provisioning system. Counters that are exposed must be first declared in a counter dictionary that is available and shared by all the SAP users and catalogs
- Counters can be shared between the charges added to a charge plan or to an offer
- Counters can be shared between the charges assigned and activated in a provider contract

Price Plan

A price plan is a set of instructions used to compute the cost of a charge. It is graphically represented in the Core Tool user interface by a tree called “decision tree”. The price plan is an essential concept of SAP CC. It represents the basic structure for charges and pricing macros business objects.



Price plan components

Price plans are made up with different components used as calculation tools for the creation of price plans or pricing macros. A combination of these components makes up a decision tree which is required to compute the cost of a charge.

Available components are:

- Rates
- Comparisons
- Operators
- Splitters
- Functions

Counters

A price plan includes the declaration of the necessary counters. The persistent counters will be created in the provider contract or subscription related to each customer.

Parameters

A parameter is a variable which is defined in a charge or in a refill logic. Each parameter has a name, a type and a default value. A parameter introduces a new numerical, date or string type property to the rating context, and gives the possibility to:

- Retrieve this new property using the calculation system
- Redefine the parameter for each service provider as part of their offer or charge plan
- Redefine the parameter for each customer as part of his business agreement (at provider contract or subscription level) with a service provider



NOTES

- The value of a parameter is initialized in a charge or in a refill logic object
- The value of a parameter can be redefined by an SAP User at the following levels:
 - In a charge plan or in a refill plan
 - In an offer
- The value of a parameter can be redefined by an external provisioning or CRM system at the following levels:
 - In a provider contract
 - In a subscription
- Some particular types of parameters are predefined and can be used when modeling a charge plan or a refill plan. The parameters that are visible from the external provisioning or CRM system can have the following types and formats:
 - Mapping table identifier (string)
 - Mapping table key (string)
 - Range table identifier (string)
 - Invoice text (string)

Pricing macros

A pricing macro is a tool which gives the possibility for users to optimize the creation and modification of price plans. A pricing macro is a small reusable element capable of retaining and applying a calculation formula. A price plan including common calculation formulas can thus be easily managed and maintained by using pricing macros.

Translation tables

In order to develop and simplify complex price plans, users sometimes need to create properties that do not exist in the rating context.

A translation table is a tool dedicated to this kind of price plans. It gives the possibility to generate new properties based on existing properties and, at the same time, search within a table for corresponding output values.



NOTE

It is not possible to override a translation table in a charge plan, in a refill plan or in a provider contract.

Tier tables

In order to model complex price plans, users sometimes need to implement tiered tariffs. To ease such implementations, a new price plan component has been introduced to manage tier tables. This component gives the possibility to define different output values for different ranges of a numeric input property.

 **EXAMPLE**

Input range	Output range		
	Value #1	Value #2	Value #3
$x \leq 60$	1	2	
$60 < x \leq 120$	2	3	A
$120 < x \leq 200$	4	4	AV
$200 < x$	6	5	GH

 **NOTES**

- Tier tables are handled in chronology (each row of a table can be modified, which leads to a new version of the table)
- To compute the output value of a numeric output column, 4 different modes have been implemented:
 - Single, non-linear
 - Single, linear
 - Cumulative, non-linear
 - Cumulative, linear
- Tier tables can be used in a price plan or in a charging plan
- Tier tables can be overridden in offer conditions, charge conditions and charge activations of subscriptions using a specific tier table whose definition thus has a higher priority
- It is not possible to override a tier table in a charge plan, in a refill plan or in a provider contract

Mapping tables and mapping table classes

A mapping table is a table of correspondence for mapping an input set of values to an output set of values defined for different periods of time. According to a reference date, SAP CC selects the right output set during charging or refilling operations.

Mapping tables can be used in a price plan, in a charging plan and in a decision tree of a refill logic. Input values of a mapping table can be strings and currencies, whereas output values can be strings, numbers and strings.

A mapping table class is a master data which defines a common structure (input and output value sets) for several mapping tables and subscriber mapping tables. Service pro

 **NOTES**

- To model flexible price plans and refill logic, the selection of the mapping table which must be used at runtime for a charging or refilling operation can be customized by using a dedicated parameter. This parameter gives the possibility to select one table from all the tables having the same class.
- No redefinition or no exception is possible
- For a particular input set (also named input key), two sets of output values cannot exist for the same time slot
- Different sets of output values can be set up for time periods that overlap. The final output set will be selected by applying the following rules:
 - Rule 1: when more than one period can be identified for a given input and a reference date, the period whose start time is the closest to considered time is selected
 - Rule 2: if two periods have the same start time, the shortest one is selected first
- In an integrated scenario with the SAP ERP/FI-CA component of the SAP Business Suite, a default mapping table class is provided to manage recurring fees. The relevant mapping tables are automatically created by the SAP CRM system when releasing a commercial product (see Price Table documentation below).

- Subscriber mapping tables are mapping tables which are assigned to a subscriber account. In that case, they do not belong to any catalog and cannot be used by several customers. They can be reused in all or some of the provider contracts belonging to the subscriber account. You can also configure the customer master data so that each provider contract has its own subscriber mapping table.

Price table

A price table is a particular mapping table that is created and maintained by the SAP CRM system. It is thus only available in an integrated scenario with the SAP ERP/FI-CA component of the SAP Business Suite.

A price table gives the possibility to:

- Manage prices and price conditions for a commercial product, directly and transparently from SAP CRM
- Reuse these prices when modeling charges with recurrent fees in SAP CC

A price table is defined by a set of 2 inputs (price key, currency) and one output number (price).



NOTES

- A price table corresponds to a particular mapping table class which is provided by default. It must be added to a unique catalog of a given service provider by using the Core Tool user interface. **This object must not be modified.** No mapping table based on this class must be created (create your own class)
- Do not use a price table in a usage rate or in a one-shot rate for a price plan
- Do not create, modify or delete a mapping table based on a price table

Range tables and range table classes

A range table is a table of business data used to map ranges of positive numbers to a set of output string or number values which can represent prices, unit prices, or other credits. Some combinatorial rules can be applied to output values in order to compute final values and supplementary values. Intelligent pricing logic is associated in order to compute values by designing basic or complex algorithms.

Using a range table is recommended to design and configure a pricing logic which includes tiered pricing.

A range table class is a master data which defines a common structure (input and output value sets) that can be used for several range tables and subscriber range tables.



NOTES

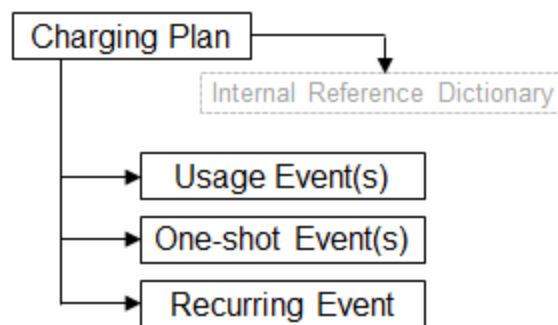
- To model a flexible pricing logic (in price plans and refill logic), the selection of the range table which must be used at runtime for a charging or refilling operation can be customized by using a dedicated parameter. This parameter gives the possibility to select one table from all the tables having the same class
- No redefinition or no exception is possible
- A particular range table named “Subscriber range table” has been designed to materialize range tables which are assigned to a subscriber account. These particular range tables do not belong to any catalog and cannot be used by several customers. They can be reused in all or some of the provider contracts belonging to the subscriber account. You can also configure the customer master data so that each provider contract has its own subscriber range table

Charging Plan

A charging plan is a set of instructions used to determine the subscriber account(s) which must be charged with the amount computed during the execution of a price plan.

NOTE

In case of charging operations based on subscriptions, a charging plan can also be used to refill the balance of a prepaid account.



Charging plan components

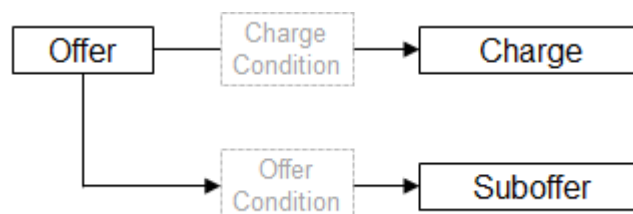
Charging plans are made up with different components used as triggering tools for account charging purposes. Like price plan components, the combination of charging plan components make up a decision tree which is used to associate usage costs and subscriber accounts by using information of the subscriptions (see charging mapping) or provider contracts (see account assignments).

Available components are:

- Charging
- References
- Comparisons
- Operators

Offer

An offer is an object which describes the way to sell one or more customer services. It corresponds to templates of charging agreement conditions (such as usage and/or recurring fees) and a validity period during which customers can subscribe to. Each condition is represented by a specific charge called charge condition.



Two different types of offer are implemented:

- Simple offers
- Complex offers, used to describe offers such as packages, options, business-to-business offers, and so on

NOTE

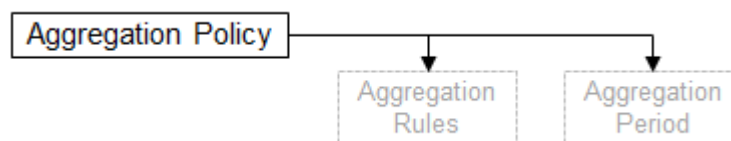
An offer always contains at least one pricing condition (charge) or another offer (called suboffer and redefining the charging conditions of the parent offer through an “offer condition”).

Aggregation Policy

An aggregation policy is a master data which gives the possibility to merge multiple charged items into a single one, during charging operations performed in a session-based execution mode only.

An aggregation policy refers to a charged item class, in order to determine the charged items that must be aggregated. It is also made up with the following information:

- A set of aggregation rules, which are associated to the different properties of the selected charged item class
- An aggregation period, which gives the possibility to schedule an automatic reporting of the pending charged item(s), in case of session inactivity for example



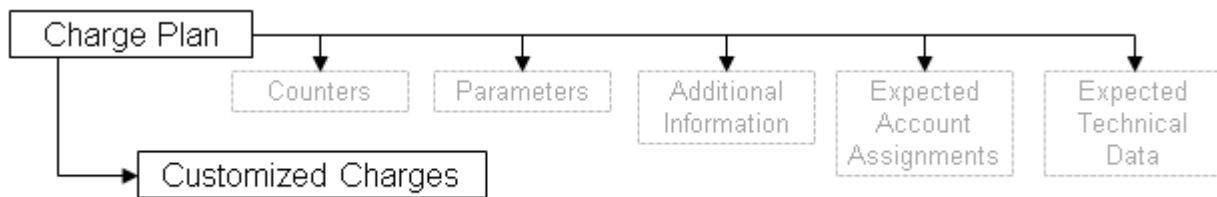
Notes

- Aggregation policies can be assigned to customized charges of charge plans, when configuring the generation of charged items resulting from charging operations. As a charge plan can contain multiple customized charges (related to master or dependent charges) which can be associated to aggregation policies, multiple pending charged items can coexist in the same session
- The Charged Items Aggregation feature only concerns charged items related to Usage events, as the other events are triggered externally during the execution of the Activation process
- Aggregation policies are not handled in chronology

Charge Plan

A charge plan is a master data which models the way to charge the access and the use of a marketable service by a given customer. It must be bundled in a commercial product in an interfaced CRM system. The charge plan is the key master data of the new model introduced in SAP CC 3.0.

A charge plan includes redefined and customized charges, redefined counters or parameters. It also includes the declarations of the elements that are expected in the provider contract for each activated charge plan (account assignments, user technical identifiers, counters and parameters).



Combination of charge plans

Once created in the system, the charge plans are combined in an external CRM system (or other provisioning system) to model commercial products or offers for a service provider. To sell these products, long terms business agreements are signed by customers. This leads to the creation of a provider contract that can be distributed to connected systems: its charging view includes activation data of charge plans and refill plan (on SAP CC side).

Note

In an integrated scenario with the SAP CRM system, a refill plan is assigned to a commercial product from the Interaction Center.

Charges and dependencies

A charge plan is made up with several configured and customized charges. Inside a charge plan, a dependent charge must be linked with either another dependent charge or a master charge to be synchronized with.

Counters and parameters

A charge plan can include some declarations of counters and parameters in order to:

- Share counters or parameters between the customized charges inserted into the charge plan
- Redefine the initial value of some counters or parameters present in a customized charge inserted into the charge plan

NOTE

In an integrated scenario with the SAP ERP/FI-CA component of the SAP Business Suite, a parameter named “SAP_BILL_CYCLE_DATE” can be used to correlate or synchronize the recurrent fees or refills with the billing cycles.

SAP CC gives the possibility to share a counter at provider contract level. The visibility level of the concerned counter must be set to “external” in the charge plan in order to let the external CRM or provisioning system manage this share.

NOTES

- This sharing function must be implemented in the integration of your CRM or provisioning system by using the technical interfaces of SAP CC 5.0 and the appropriate configuration of your charge plans
- This sharing function can be customized in order to use namespaces to restrict the sharing in the provider contracts
- In an integrated scenario with the SAP CRM system, the combination of charge plans is directly possible from the Interaction Center

SAP CC provides a visibility function in order to let the external CRM or provisioning system redefine the value of parameters. A charge plan must include the parameters which are visible by these systems.

 **NOTE**

The “internal” visibility is used to share counters or parameters between the different charges inserted and customized in a charge plan.

User Technical Identifiers

A charge plan includes the declaration of the user technical identifiers that will be expected in the activation of this charge plan (see contract item). A User Technical Identifier (UTI) corresponds to a value which identifies a customer for several marketable services. For example an e-mail address, a phone number, the IMSI or the MSISDN can be used as a UTI for several customer services such as SMS, MMS, voice calls or data transfers.

 **NOTE**

For each charge inserted into a charge plan and including a price plan with a usage rate, it is necessary to attribute the service identifiers to these UTIs.

Expected Account Assignments

A charge plan includes the declaration of the account assignments that will be expected in the activation of this charge plan (see contract item). This declaration is used by the external CRM or provisioning system to require the necessary accounts when creating a provider contract during a sales order.

An account assignment is an attribution of a prepaid or external postpaid account to a contract item of a provider contract. It is used during the charging and refilling operations to determine the account(s) which must be credited or debited.

Each contract item includes all the mandatory account assignments as originally declared in the activated charge plans.

Versions

Different versions of a charge plan can coexist in the system for different periods of time.

Lifecycle

A lifecycle management function based on a status is available for a charge plan in order to restrict its use once it is set to the “released” status.

Counter name dictionary

A list of counter names and descriptions is available and mandatory in case of counters declared in a charge plan or a refill plan and that will be visible in an external CRM or provisioning system. This dictionary is available for all the SAP Users and for all the pricing catalogs in the SAP CC system.

Customized Charge

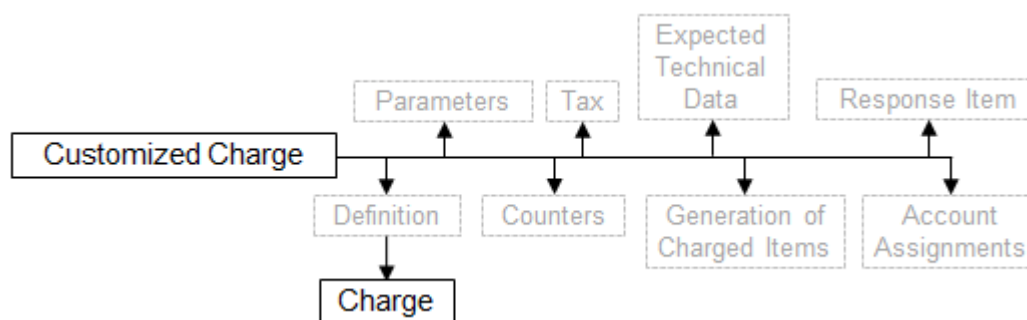
A customized charge is a reusable charge inserted in a charge plan when modeling the pricing of customer services. It includes the characteristics of this charge and some possible customizations such as:

- The redefinitions of persistent counters

- The redefinitions of parameters
- The configuration of the tax relating to the consumption of the customer service
- The configuration of the generation of the charged items data files
- The configuration of the details included in the response of the operation execution
- The mapping between the charging references (of the related charge) and the expected account assignments (of the charge plan)
- The configuration of the service identifiers

 **NOTE**

A customized charge inserted into a charge plan is technically called a “charge plan item”. For further information about charge plan items, refer to the [SAP Convergent Charging Web Services documentation](#).



Counters

The initial values of persistent counters can be redefined by linking these counters to new counters declared in a charge plan.

Parameters

The value of each parameter originally set up in a reusable charge can be redefined if necessary in the customized charge.

It is also possible to associate a parameter to a new parameter declared in the charge plan in order to let the external CRM system redefine the value of the parameter when creating a provider contract for a customer of the service provider.



Tax Settings

By default, a customized charge inherits from the tax settings defined in the reusable charge. Nevertheless, it is possible to override these tax settings by defining some specific tax settings to a customized charge.

The following invoicing tax systems are supported:

- EU Value Added Tax (EU VAT), which gives the possibility to specify a unique tax level and a VAT rule
- Flat Tax, which gives the possibility to specify a specific tax rate value
- U.S. Telco Tax, which uses the BillSoft EZTax third-party tax system

System	Description
--------	-------------

EU VAT	<p>Price plans output amounts which are technically named “Base Amount” and which correspond to net amounts by default. Referring to price plans, the customized charges give the possibility to override this default configuration and specify if a “Base Amount” must be a net or gross amount.</p> <p>Using the VAT system consists in specifying a given tax level, which can be:</p> <ul style="list-style-type: none"> ■ A predefined tax level such as normal, reduced, super reduced, and so on ■ A level coming from a property of the rating context <p>The tax rate associated to a given tax level may vary between countries (e.g. “normal” level in France corresponds to 19,6%, but corresponds to 19% in Germany). It is thus necessary to customize SAP CC with all the tax rates and country codes that are necessary to represent the tax information for the end customers of a service provider.</p> <p>The correspondence between countries, tax levels and associated tax rates is defined in a dedicated file named “VATRate.txt”. The content of this file is recorded in both database and memory for performance reasons.</p> <p> EXAMPLE</p> <p>The following text represents the “normal” tax rate for France: <code>/FR/0; France VAT; 0.196; 01/05/01; 0</code></p> <p> CAUTION</p> <ul style="list-style-type: none"> ■ When a customized charge is configured to handle “net” amounts, all the amounts handled by the associated price plan are assumed to be net amounts. Otherwise, taxes are computed twice. ■ When a customized charge is configured to handle “gross” amounts, all the amounts handled by the associated price plan are assumed to be gross amounts. Otherwise, no tax can be computed. <p>A VAT rule can be selected to support the 2010 EU VAT Invoicing Directive to determine the places of taxation according to the business requirements.</p> <p>Two main use cases are B2B and B2C commercial exchanges.</p>
Flat Tax	<p>Price plans output amounts which are technically named “Base Amount” and correspond to net amounts by default. Referring to price plans, the customized charges give the possibility to override this default configuration and specify if a “Base Amount” must be a net or gross amount.</p> <p>Using the flat tax system consists in specifying a given flat tax rate value, which can be:</p> <ul style="list-style-type: none"> ■ A constant value ■ A value coming from a property of the rating context
U.S. Telco Tax	<p>The U.S. Telco Tax system is built upon the BillSoft EZTax third-party tax system, which must be acquired separately to SAP CC. This system requires a large number of parameters which are specific to U.S. Telco taxes, such as:</p> <ul style="list-style-type: none"> ■ The origin of a call ■ The destination of a call ■ The type of service which must be charged ■ The type of customer (B2B/B2C)

- And so on

For further information about the U.S. Telco Tax system, refer to the documentation of the BillSoft EZTax third-party tax system in order to correctly set-up your charge according to your business.

Configuration of the generation of charged items

A customized charge includes the configuration of the generation of the charged items resulting from charging operations. These output records can include various details computed by SAP CC when performing charging operations at run-time.

In case of charging operations performed in session-based execution mode, it is also possible to specify an aggregation policy in order to aggregate the charged items before outputting them, as well as a reporting scope to configure which charged items are reported when a reporting condition from an aggregation policy is fulfilled.

Configuration of the generation of response items

A customized charge includes the configuration of the details included in the response of the operation execution. The returned transaction details include a filtered list of properties computed during the execution of a price plan. If no response item is configured, all computed transaction details are returned.

Charging reference mapping

Each charging reference declared and used in the charge of the customized charge must be mapped to one of the expected account assignment declared in the charge plan.

Service ID declaration

A customized charge must include the technical identifiers of the customer services that can be charged by SAP Convergent Charging for their usage. Each identifier must be associated to a UTI declared in the charge plan.



NOTES

- These technical identifiers are named Service IDs and must be present in the chargeable items received by SAP CC from the mediation system as transactional data
- These declarations are not relevant if the customized charge only includes recurrent or one-shot rates in its price plan

Refill Item

Refill items correspond to **input** information processed by SAP CC when receiving a refill request. A refill item contains the following information:

- The different properties which are used to define a type of refill
- The identifiers of a customer, which are used to credit a prepaid account

Refill item class

A refill item class is an assembly of characteristics that can be associated to a refill item. Refill item classes must be configured in order to model the refill logic for usage refills. The algorithms can be based on the properties declared in the class.

Refill Record

Refill records represent **output** information generated by SAP CC when processing a refill request. For example, a refill record can contain some catalog data, the origin of the refill, some properties of the refill, some computed properties, and some prepaid account information.

Refill record class

A refill record class is an assembly of characteristics that can be associated to a refill record. Refill record classes can be optionally configured and reused when customizing the refill logic inserted in a refill plan.



NOTES

- Taxes do not apply to refill record classes
- In an integrated scenario with the SAP ERP/FI-CA component of the SAP Business Suite, the configuration of the refill record classes is simplified and automated by creating the billable item mapping

Refill Logic

A refill logic is a reusable object which defines the calculation algorithms for different credits (money, units, and so on) of a refill operation. It can model an algorithm for manual refills, automated periodic refills, one-shot refills and account event refills.

Every refill logic is customized within a refill plan.

Components

The refill logic includes some decision trees that consist of a set of instructions based on refill components which are similar to the components of a price plan.

Available components are:

- Refills
- Functions
- Comparators
- Splitters
- Operators

The refill logic can include pricing macros to share and reuse parts of instructions.

Persistent counters

The refill logic includes the declaration of the necessary counters. The persistent counters will be created in the provider contract of each customer.

Parameters

The refill logic can include some parameters. The default value can be redefined in a refill plan or in a contract item referencing this refill plan.

Refill Plan

A refill plan defines generic terms of refilling used to credit the prepaid account of a customer. Refill plans are necessary if you want to provide the end customers with refill services in addition to the marketable services.

A refill plan contains a refill logic and some configurations.

Combination

Once released in the system, a refill plan can be bundled with some charge plans in a CRM system or an external provisioning system in order to model commercial products and offers. Then it is activated in the provider contract of the customer when he subscribes to these products.



NOTES

- Taxes do not apply to refill plans
- In an integrated scenario with the SAP CRM system, a refill plan is assigned to a product from the Interaction Center

Counters and parameters

A refill plan can include:

- Some parameters used to redefine the default value of some parameters present in the customized refill logic
- Some declarations of counters in order to redefine the initial values of some counters present in the customized refill logic



NOTE

In an integrated scenario with the SAP ERP/FI-CA component of the SAP Business Suite, a parameter named "SAP_BILL_CYCLE_DATE" can be used to correlate or synchronize the recurrent refills with the billing cycles.

Counter sharing at provider contract level

SAP CC provides gives the possibility to share a counter at provider contract level for all the charges and the refill logic available in the different contract items. The external CRM or provisioning system can manage the share.



NOTES

- This function must be implemented in the integration of your CRM or provisioning system by using the technical interfaces of SAP CC 5.0 and the appropriate configuration of your charge plans and a refill plan
- This function can be customized by using namespaces to restrict the sharing in the provider contracts

Parameter redefinition at provider contract level

The default value of the parameters defined in a refill plan can be redefined by the external CRM or provisioning system when creating a provider contract and adding a contract item referencing this refill plan.

User Technical Identifiers

A refill plan optionally includes the declaration of the user technical identifiers that will be expected in the activation of this refill plan (see contract item). A User Technical Identifier (UTI) is a single key that identifies a customer for several marketable services and for different refill services. For example an e-mail address, a phone number, the IMSI or the MSISDN can be used as a UTI for several customer services (SMS, MMS, voice, data).



NOTE

The configuration of the User Technical Identifiers depends on the integration of SAP CC with external systems that can deal with refill requests.

Expected account assignment

A refill plan includes the declaration of a unique account assignment that is expected in the **activation** of this refill plan in a provider contract. This declaration is used by the external CRM or provisioning system to get the necessary prepaid account when creating a provider contract during a sales order.

Versions

Different versions of a refill plan can coexist in the system for different periods of time.

Lifecycle

A lifecycle management function based on a status is available for a refill plan in order to restrict its use once it is set to the “released” status.

Counter name dictionary

A list of counter names and descriptions is available and mandatory in case of counters declared in a charge plan or a refill plan and that will be visible in an external CRM or provisioning system.

This dictionary is available for all the SAP Users and for all the catalogs in the system.

Customized Refill Logic

The customized refill logic is a reusable refill logic object inserted in a refill plan when modeling the pricing of customer services that features refilling functions for prepaid accounts. It includes the characteristics of this refill logic and some possible customizations such as:

- Redefinitions of persistent counters
- Redefinitions of parameters
- Configuration of the generation of the refill records (generated by SAP CC)
- Attribution of the charging reference with the **expected** account assignment
- Configuration of the service identifiers

NOTE

A customized refill logic inserted in a refill plan is technically named refill plan item. For further information about refill plan items, refer to the [SAP Convergent Charging Web Service](#) documentation.

Counters

The initial values of persistent counters can be redefined by linking these counters to new ones declared in a refill plan.

Parameters

The value of each parameter originally set up in a reusable refill logic object can be redefined in the customized refill logic added to a refill plan. It is also possible to associate a parameter to a new parameter declared in the refill plan in order to let the external CRM system redefine the value of the parameter when creating a provider contract for a customer of the service provider.

Configuration of the generation of the refill records

The customized refill logic includes the configuration of the generation of data (refill record) resulting from the refilling function and that must be taken into account by an external billing system. These output records can include various details calculated by SAP CC when performing the refilling function at run-time.

Charging reference attributions

The unique charging reference declared and used in the refill logic must be associated with the account assignment expectation declared in the refill plan.

Service ID declaration

The customized refill logic must include the technical identifiers of the refill services which can be used by SAP CC to identify the service which is supposed to handle the refill requests. Each of these technical identifiers must be associated with a UTI declared in the charge plan in order to identify the concerned prepaid account.



NOTES

- These technical identifiers are named Service IDs and must be present in the chargeable items received by SAP CC from the mediation system as transactional data
- These declarations are not relevant if the customized charge includes only recurrent or one-shot rates in its price plan

Monitoring Plan

A monitoring plan defines the generation of spending statuses that you may want to monitor in a connected policy system (or monitoring system) during charging and refilling operations relating to an end customer. The plan defines a list of spending status identifiers reported by SAP CC. For each declared spending status, the plan defines the monitored source (counter in contracts) and some logic settings for defining some label values for the reported spending status and the spending thresholds that relate to these spending status labels.

The monitoring plans are then combined and assigned to commercial products in a CRM or provisioning system and finally activated in the provider contract of an end customer of the service provider.

The monitoring plan consists of:

- One or more configurations of target spending statuses to be generated and reported by the SAP CC system
- A set of parameters to configure the monitoring plan; the default value is the name of a range table.
- A release status which can be open, released, or obsolete



NOTES

- You create monitoring plans only when SAP CC is connected to a policy control system and when the spending status monitoring function is implemented in your system landscape.
- SAP CC manages monitoring plans to configure the spending status monitoring feature in master data. When SAP CC is connected to SAP CRM, you must enable a switch in SAP CC so that SAP CRM can consider the monitoring plans as charge plans.

Spending threshold settings

You configure the logic to convert the values of a counter (in a contract) to a label reported in a spending status by setting up a range table dedicated to this function. You configure a range table for each spending status that you want to monitor in the policy system.

NOTES

- You need to configure particular range tables that are defined by a unique output column as type of string. The upper bounds of ranges define the spending thresholds (or spending limits) applicable to the corresponding counter values.
- When a counter value is out of these ranges, the SAP CC system considers that the spending status label is unknown.
- You must set up the relevant range table class in the pricing catalog of the service provider. You can set the range upper bounds to Inclusive or Exclusive depending on the business requirements. If the last range upper bound is limited the SAP CC system considers the spending status.

Versions

Different versions of a monitoring plan can coexist in the system for different periods of time.

Lifecycle

A lifecycle management function based on a release status is available for a monitoring plan to restrict its use between different statuses.

Counter name dictionary

A list of counter names and descriptions is available and mandatory in case of counters declared in a monitoring plan or a refill plan and that will be visible in an external CRM or provisioning system. This dictionary is available for all the SAP Users and for all the catalogs in the system.

Spending status descriptions

A list of possible spending statuses and their label description is available the SAP CC system. This configuration must be known in the downstream system.

Allowance Event

The allowance event is a technical transactional data record which contains the list of properties (and their associated values) required to use allowances.

In the decision trees of allowance logic objects, refill logic objects or price plans, an allowance event corresponds to a message which is sent to a configured set of allowances of the same provider contract, informing for example, that a quantity of service is consumed.

Processing an allowance event may consist in:

- Modifying the allowance itself (e.g. counters and/or validity period)
- Updating the allowance event if necessary

EXAMPLE

A message is sent to a set of allowances of a provider contract.

- One of these allowances extends its validity period
- Another allowance debits a quantity of service by modifying the allowance event



NOTES

- The allowance that sends an event does not receive it
- Allowances cannot react to an event by sending a new event

For more information on how to send an allowance event, read the chapter about the Allowance Event Sender Operator component, available in the [SAP Convergent Charging Logic Components Reference](#) documentation. For more information about how to configure an allowance for listening to a given allowance event, read the chapter about the Allowance Logic and the Allowance Plan, available in the [SAP Convergent Charging Core Tool Online Help](#) documentation.

Allowance Event Class

An allowance event class contains a list of property descriptions which define the type of a given Allowance Event. Each property description defines the name and the type of a property.



NOTE

The name of the allowance event class defines the type of the event (e.g. an allowance event class named "ALLOWANCE_DEBIT" defines the event associated to the debit of an allowance).

Allowance Logic

An allowance logic object:

- Defines the rules and conditions of use of an allowance
- Can model an algorithm for several event-based services, one-shot, and recurrent allowance events
- Is shareable, or not



NOTE

The allowance logic can be of two types:

- **Shareable:** This type of allowance logic gives the possibility to create shared allowances from shareable allowance plans. Such configuration implies some restrictions regarding the use of logic components (for further information, refer to the [SAP Convergent Charging Logic Components Reference](#) documentation). This allowance logic can also be used to create allowances that are non-shared
- **Not shareable:** This allowance logic is only used to create allowances that are accessed within a subscriber account

The calculation algorithms can use properties relating to allowance events described in different allowance event classes. An allowance plan gives the possibility to customize any allowance logic object.

Components

An allowance logic object can use the following components:

- Functions

- Comparators
- Operators

 **NOTE**

An allowance logic object can also include pricing macros to share and reuse parts of instructions.

Persistent counters

An allowance logic object includes the declaration of all the necessary persistent counters. You can create persistent counters in the allowances of a provider contract.

 **NOTES**

- The allowance logic uses a set of persistent counters that are stored in the database for modeling allowances. In the case of shared allowances, these persistent counters are considered as distributed counters. Such configuration implies some constraints regarding the usage of these distributed counters in the shareable allowance logic. For example, only the debit of a positive value (Debit Counter Operator component) from a distributed counter is allowed in the shareable allowance logic. A distributed counter cannot be read, and thus, it is not visible in the shareable allowance logic. For more information about the Debit Counter Operator component, refer to the [SAP Convergent Charging Logic Components Reference](#) documentation.
- If the shareable allowance logic is used by allowances that are not shared, persistent counters are created and defined in the same way as in charge or refill plans. For example, these counters are instantiated only for a subscriber account and are visible and used by the provider contracts that belong to this subscriber account.

Transient counters

An allowance logic object includes the declaration of all the necessary transient counters.

Parameters

An allowance logic object can include some parameters. You can redefine any default value defined in a customized allowance logic object referencing this allowance logic. See below for further information about customized allowance logic objects.

Customized Allowance Logic

A customized allowance logic object is a reusable allowance logic inserted in an allowance plan for modeling the pricing of customer services. A customized allowance logic object is technically named allowance plan item.

An allowance plan item includes the characteristics of the allowance logic and can be customized to:

- Redefine persistent counters
- Redefine parameters
- Configure the generation of charged items (output data generated by the SAP CC system)

Allowance Plan

An allowance plan is a master data which models the behavior of an allowance at the time of its creation, activation, usage, and expiration.

 **NOTE**

The allowance plan can be of two types:

- **Shareable:** This type of allowance plan is used to create allowances that are shared between subscriber accounts. However, such allowance plan also gives the possibility to create allowances that are not shared. This allowance plan only consists of shareable allowance logic objects
- **Not shareable:** This allowance plan is only used to create allowances that are accessed within a subscriber account. This allowance plan consists of allowance logic objects that are shareable or not shareable

Life cycle

During its life cycle, an allowance plan has one of the following statuses:

- **Open:** This status is automatically assigned to the allowance plan at creation time. This status:
 - Gives the possibility to modify the allowance plan
 - Does not allow creating an allowance using the Create Allowance component in a charge plan, in a refill plan, and in an allowance plan. A Create Allowance component must reference a released allowance plan for creating an allowance.
- **Released:** This status only gives the possibility to make few changes to the allowance plan. It also gives the possibility to create an allowance using the Create Allowance component, which stores the newly created allowance in the provider contract of the customer

Versions

Different versions of an allowance plan can coexist in the system for different periods of time.

Allowance Interface

An allowance interface lists and describes the common parameters and counters used by all the allowances. It gives the possibility to extend the default set of properties of allowances including information such as the unique identifier, the activation date (Validity Start Date) or the expiration date (Validity End Date).

 **NOTE**

The persistent counters that are defined in the allowance interface cannot be read and are not visible if they are implemented by shared allowances

 **EXAMPLE**

You can add parameters or counters to the Allowance Interface to:

- Sort allowances in the allowance event sender operator component
- Compute properties in the allowance property introducer operator component

A Change List is a master data which is only relevant when implementing the Catalog Transport feature. This master data contains a set of versions of other master data which must be transported together to a remote SAP CC system.

A change list is a master data including master data such as:

- Offer
- Pricing Macro
- Translation Table

- Tier Table
- Mapping Table Class
- Mapping Table
- Range Table Class
- Range Table
- Chargeable Item Class
- Charged Item Class
- Charged Item Aggregation Policy
- Charge
- Charge Plan
- Refill Item Class
- Refill Record Class
- Refill Logic
- Refill Plan
- Allowance Event Class
- Allowance Logic
- Allowance Plan
- Monitoring Plan

**NOTE**

A change list contains all the versions of the objects. Any change to these versions after creating the change list has no impact on its content.

A change list can also include dependent objects such as:

- Billable Item Mapping
- Consumption Item Mapping
- Allowance interface
- Counter Name Dictionary
- Refill Record Mapping
- Spending Status Description

**NOTE**

Dependent objects can be included in a change list automatically. For example, adding a charge plan to a change list may include the charges, the mapping tables, the range tables, and the other objects directly and indirectly referenced by this charge plan.

Master Data for the Customer

Access

An access corresponds to a technical key under which a customer consumes a marketable service. This identifier gives SAP CC the possibility to identify the related charges required for rating and charging purposes according to a reference date.

For a subscription containing one or more charges, you can create multiple accesses for each master charge. Creating more than one access for the same customer allows this customer to use a service under different identities.

For a provider contract, SAP CC automatically creates and maintains all the necessary accesses that are relevant for each contract item of a provider contract.

 **NOTE**

An access is technically linked to a charge through an accessible charge, and thus represents a unique way to access to a given charge activation for a certain period of time.

Subscriber Account

Before consuming services related to commercial products and offers they subscribed to, customers must be associated to accounts. These accounts are called subscriber accounts, and are used to charge the service consumptions. They are made up of different information such as:

- Tax settings (tax system, taxation mode, and so on) available for each prepaid account or postpaid account of the subscriber account
- Additional information
- One or more prepaid accounts
- A list of external accounts, which references postpaid accounts located outside SAP CC
- One or more credit-limit balances, which extends the flexibility of balances management


 **NOTES**

- Like any other account balance, balances of prepaid and external accounts are debited every time customers consume a service and can be credited through refill operations
- The balances which must be impacted is determined through the charging plan which represents the association between a charge (calculated through a price plan) and an account balance
- Credit-limit balances are not available in case of convergent charging services based on provider contracts
- In a system landscape containing several SAP ERP/FI-CA systems, each subscriber account includes information about the customer management area it belongs to. In this area, an SAP ERP system and an SAP CRM are responsible for:
 - A subset of the master data
 - The related transactional data (billable items and consumption items)
 - The possible notifications (empty prepaid balances, expired prepaid accounts)

Tax settings

A subscriber account contains tax settings which is shared with all its related accounts (prepaid and external postpaid). By default a new subscriber account is assumed to use the EU VAT system, and the service provider is assumed to pay the tax amount to the tax authority. Nevertheless, it is possible to set up these tax settings by defining the following specific options:

Tax Settings	Description
--------------	-------------

Tax system	<p>The following invoicing tax systems are supported:</p> <ul style="list-style-type: none"> ■ None, which gives the possibility to deactivate the tax calculation process for the related account ■ Both VAT and Flat Tax, which gives the possibility to specify a unique tax level and VAT rule in the tax settings of the customized charges ■ U.S. Telco Tax, which uses the BillSoft EZTax third-party system <p> NOTE</p> <p>For further information about the tax systems, refer to the definition of the customized charge master data located in the “Master Data for the Service Provider” section of this documentation.</p>
Taxation mode	<p>The taxation mode gives the possibility to specify who is supposed to pay the computed tax amounts to the tax authority. The following taxation modes are supported:</p> <ul style="list-style-type: none"> ■ Service Provider Subject to Pay, which must be used when the service provider pays the tax amounts to the tax authority ■ Account Holder Subject to Pay, which must be used when the customer pays the tax amounts to the tax authority ■ Account Holder Exempted, which must be used when the customer is granted with a tax exemption. Note that even if they are not applied on the “Base Amount” property of the price plans, tax amounts are computed for reporting purposes ■ None, which gives the possibility to disable the tax calculation process. Note that this option also avoids any reporting related to tax
Tax specific settings	<p>The tax specific settings depend on the selected tax system, and correspond to specific information that may or must be filled to compute the tax amounts.</p> <p>When the Value Added Tax system is selected, a taxation location named <i>Tax Zone</i> must be specified. This option can be used to specify:</p> <ul style="list-style-type: none"> ■ That no tax applies to the accounts (prepaid and postpaid) of the subscriber account ■ That the VAT rule specified in the charge customized in a charge plan applies if any. If no rule is specified in this charge, no tax applies to the accounts (prepaid and postpaid) of the subscriber account ■ The ISO 3166 country code of the taxation place of the holders of prepaid or external postpaid accounts in the subscriber account. This location gives the possibility to determine in which country the tax amounts are computed and thus to retrieve a list of possible tax levels and associated rates <p>When the U.S. Telco Tax system is selected, the following information can be set:</p> <ul style="list-style-type: none"> ■ Origination, which correspond to the origin of the event (address, geographical code, and so on) ■ Termination, which corresponds to the destination of the event (address, geographical code, and so on) ■ Service Address, which corresponds to the location of the service provider (address, geographical code, and so on) ■ Customer Type, which gives the possibility to specify if the customer is a private customer or a business customer

- **Incorporated Code**, which gives the possibility to specify if the customer lives inside or outside of an incorporated area (which corresponds to a geographical zone)
- **Resale Flag**, which gives the possibility to specify if the customer is a seller or a reseller

**NOTE**

The U.S. Telco Tax specific settings are optional as they can be also set in the charge customized in an offer or in a charge plan. If this information is defined in the subscriber account, it overrides the information of the customized charge.

Prepaid balance

A prepaid balance is a balance dedicated to prepaid services consumption. It represents a pre-defined available amount of money customers can use. This balance is checked every time a customer consumes a service, in order to accept or deny the request. The “Balance Management” functional module provides mechanisms to update this kind of balance and send notifications to customers.

An “Empty Limit” property has been implemented to represent the theoretical threshold under which no service consumption is allowed. When this limit is reached, customers must credit their account in order to increase the related balance.

**NOTE**

To increase the flexibility of prepaid balances management, additional mechanisms are implemented:

- **Overspending**, which gives the possibility for a customer to consume services even if the empty limit is reached
- **Overrun**, which gives the possibility to distribute the excess consumption among multiples accounts. When overrun is enabled, a consumption event can be totally or partially charged to a prepaid balance or an external balance, limiting the rejection of consumption. Note that this mechanism is not available in case of convergent charging services based on provider contracts

External account

An external account represents a link to a postpaid account managed by a third-party billing system (outside SAP CC) such as the SAP Convergent Invoicing system.

Credit-limit balance

Credit-limit balances give the possibility to refine the way services must be consumed. They behave as virtual balances associated to services and represent consumption limits that customers cannot exceed.

**EXAMPLE**

A customer subscribes to a 25\$ prepaid offer containing phone calls and SMS, but wants to limit his SMS service to 10\$. To implement this specific requirement, a credit-limit balance (attached to his subscriber account) is set to 10\$ and associated to the SMS service.

**NOTE**

The credit-limit balance management function is not available in case of convergent charging services based on provider contracts.

Alerts and notifications

Account balances can be connected to a multicast notification mechanism to inform another system (including customers) that the available amount of a balance has reached (or exceeded) a customized specific threshold. To inform customers, these notifications can be sent in real-time mode (for example, when customers are making phone calls).

Business Data Tables

Subscriber mapping tables

A subscriber account may contain subscriber mapping tables. Subscriber mapping tables are mapping tables which are attached to a subscriber account. Contrary to mapping tables, they do not belong to any catalog. Subscriber mapping tables contain subscriber specific data and can be used in the provider contracts related to the subscriber account they are attached to.



NOTE

The subscriber mapping tables are not available in the subscriptions.

Subscriber range tables

A subscriber range table is a range table assigned to a subscriber account to be shared by all or some of the provider contracts related to this subscriber account.

The subscriber range tables are recommended to design and configure the pricing logic that includes tiered pricing that must be based on customer data.

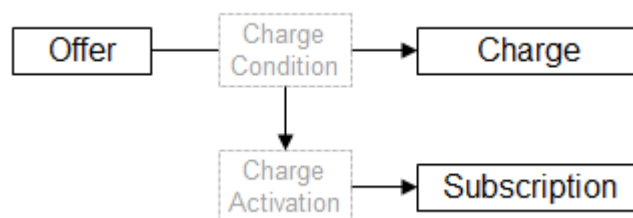


NOTE

The subscriber range tables are not available when using subscriptions.

Subscription

Subscriptions correspond to charging agreement conditions subscribed by customers and taking into account their specific needs. It thus represents the customized version of a given offer.



Like their associated offers, subscriptions are multi-level in terms of organization. However, a subscription does not necessary subscribe to all charges belonging to the offer. Only the mandatory charges are subscribed. For the other ones, users select the charge they want to subscribe to. When a charge condition is subscribed from the offer, a charge activation is technically created inside the subscription.

Three different types of subscriptions are implemented:

- **Online** subscriptions, which are charged in real-time mode and which are not compatible with the Chargeable Items Recharging process
- **Offline** subscriptions, which are always charged in a batch mode and which generate counter snapshots required for rerating purposes
- **Hybrid** subscriptions, which are charged in a batch or real-time mode according to the consumed service, which are unable to generate counter snapshots, and which are not compatible with the Chargeable Items Rerating process

Process	Online	Hybrid	Offline
Chargeable Items Charging (real-time execution mode)	■	■	
Chargeable Items Charging (batch execution mode)		■	■
Chargeable Items Rerating			■

Provider Contract

A provider contract corresponds to pricing and charging agreement conditions approved by an end customer with a service provider and taking into account his specific conditions. In SAP CC, a provider contract refers to a unique subscriber account, and contains all the elements which are necessary to execute charging and refilling operations.

Provider contracts are created by an external CRM or provisioning system when capturing sales orders or contract signatures.

NOTE

In a solution scenario with the SAP CRM or SAP ERP/FI-CA components of the SAP Business Suite, a provider contract is a cross-component master data. Each system (SAP CRM, SAP CC, and SAP ERP) manages its specific part of a provider contract, and exchanges the other parts with the other components. Some replication and distribution mechanisms are available to manage these business objects.

Contract item

A provider contract includes multiple contract items. Each contract item corresponds to an activation of a charge plan or a refill plan originally set up in SAP CC and which is part of the master data for the service provider.

Activated plan (charge plans and a refill plan)

When modeling the commercial product, the external CRM or provisioning system is responsible of the consistency between subscriber accounts, provider contracts and the combinations of activated plans. A given provider contract can activate multiple charge plans (or multiple times the same charge plan), but **can only activate a unique refill plan**. Moreover, all the activated refill plans related to a given subscriber account must assign a different prepaid account (a prepaid account cannot be assigned in more than one activated refill plans related to a given subscriber account).

Shared counter

A provider contract can include some counters which can be shared between multiple contract items for charging operations. These contract items have the same namespace.

To model enterprise credit pooling, the counter share can be extended between multiple provider contracts belonging to the same subscriber account. A parent contract shares some counters with some linked provider contracts.

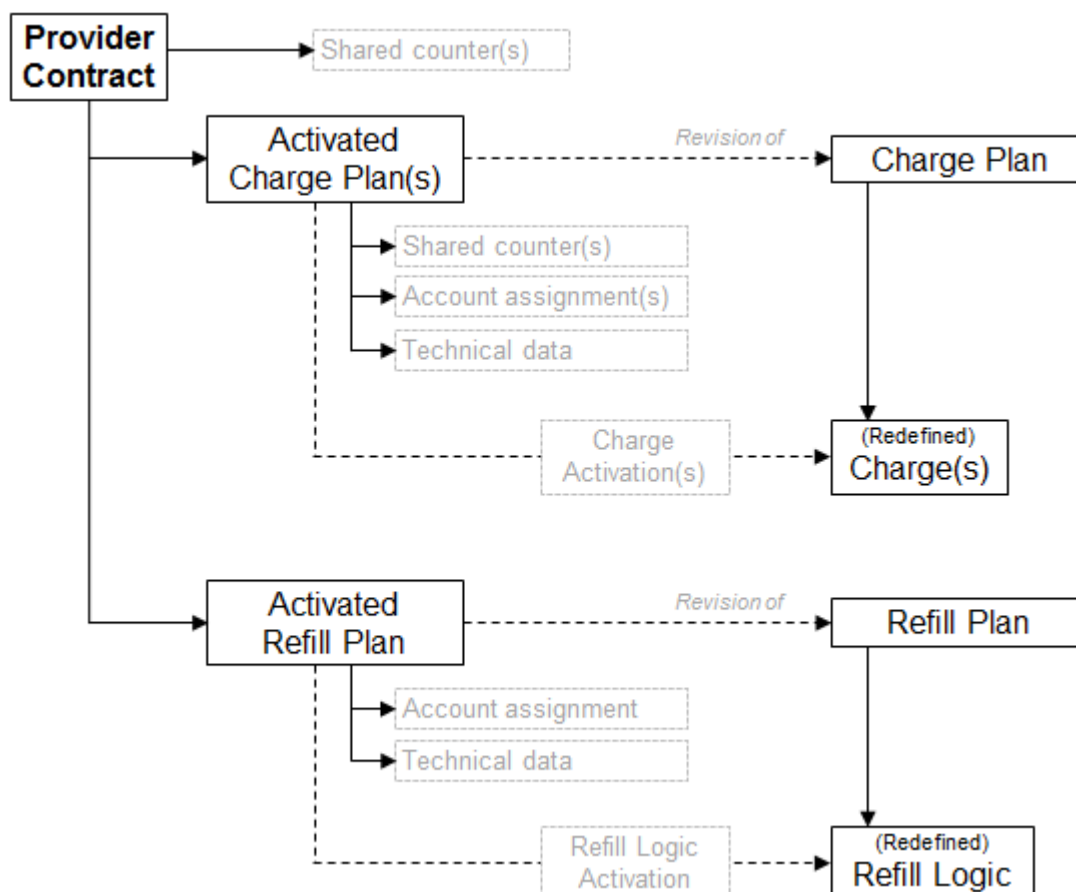
EXAMPLE OF ENTERPRISE CREDIT POOLING

A small company wants to share some pooled minutes with its employees to lower the cost of the phone bill.

To implement this pricing model in SAP Convergent Charging, you must configure a parent provider contract that shares a counter to linked contracts. The parent contract activates a charge plan responsible for crediting this counter. The linked contracts activate charge plans that debit this counter.

NOTE

The initial value of a shared counter is 0. This value can be customized if a particular charge plan activated in a contract item includes a charge with a one-shot rate which sets up the initial value of the shared counter at the creation of the contract item.



Version

SAP CC can manage different versions of a provider contract. The structure of a provider contract can change (e.g. new/removed elements), but the content of the elements it contains can also change. A revision thus represents the version of the provider contract defined for certain period of time.

These periods of time are not defined at the provider contract level, but they are defined at a sublevel: each element of the contract (activations of charge plans or of a refill plan) can be defined for a time slot. This includes the account assignment data, technical data, shared counters and parameters.

Contract Item

A contract item is the representation of the business agreement restricted to the part of the commercial product that contains a charge plan, a refill plan, or a monitoring plan defined in the pricing catalog. It refers to this plan that has been activated by the CRM or another external provisioning system when creating or maintaining the provider contract. A contract item includes the necessary counters or refers to some counters shared between multiple contract items in the provider contract or in linked provider contracts.

A contract item also includes the account assignments and the necessary technical data.

If a charge plan has been activated several times, the provider contract includes a contract item for each activation.

Charge activation

A contract item includes a charge activation for each customized charge declared in the corresponding activated charge plan. Each charge activation includes the persistent counters declared at charge level.

Counter and counter share

Each contract item includes its persistent counters. The counter can be shared with other contract items in the provider contract or with other contract items in linked contracts. Several counter namespaces are possible to restrict the counter share.

Each contract item refers to the counters shared by the other contract items with the same namespace.

Account assignment

An account assignment is an attribution of a prepaid or an external postpaid account to an element of the provider contract. It is used during charging and refilling operations to determine the account(s) to be credited or debited.

Each contract item of a provider contract includes all the mandatory account assignments as originally declared in the activated charge plan.

Technical data and user technical identifiers

Each contract item includes User Technical Identifiers (UTI) which are used by mediation systems as technical identifiers for the customers of the marketable services. These identifiers are reused by SAP CC to assign service consumption information with the real consumers. The same User Technical Identifier can be used for different services (ex.: a mobile phone number is used for SMS and MMS technical identifications).



NOTES

- If the same charge plan in the pricing catalog is activated several contract items, the USID must be different in the technical data.
- If a provider contract includes several contract items activating different monitoring plans, the original monitoring plans must not include the generation settings for the same spending statuses.

SAP CC creates and manages automatically the necessary accesses for:

- All the marketable services declared in the different charges of an activated charge plan
- All the user technical identifiers
- All the contract items of the providers contracts

Allowance

An allowance is a master data which:

- Gives the possibility for a customer to consume one or several services in a limited quantity or time
- Can be used to model a credit bucket (e.g. 3GB of GPRS data, within a 30 days validity period)
- Is linked to an allowance plan at creation time



NOTE

The validity period of an allowance ranges from its start date to its end date.

Charge activation

An allowance includes the charge activation for each customized allowance logic object declared in the corresponding activated allowance plan. Every charge activation includes the persistent counters declared at allowance logic level.

Account Assignment

An allowance relates to one of the accounts of the subscriber account relating to the associated provider contract. When an allowance is created, this account is assigned to an allowance according to the following rules:

- If the allowance is created inside a refill logic, the account corresponds to the prepaid account to be refilled
- If the allowance is created inside a charge, the account corresponds to the prepaid account or external account computed by the charging plan of this charge
- If the allowance is created in the allowance logic, the account is the account of the parent allowance (the first allowance that was created in the related charge plan or in the related refill plan)

Currency

The currency of an allowance is defined at allowance creation time. This currency is inherited from the refill logic or the charge that has created this allowance.

Allowance Validity Period

The validity period of an allowance corresponds to the period of time during which the allowance can be debited by the [Chargeable Items Charging](#) process or activated by the [Activation](#) process. It is also used to define the time at which the allowance:

- Can be informed about internal events (through allowance events)
- Can be purged by the allowances purge technical mechanism

Counters

Each allowance includes its own persistent counters. Refer to [Allowance Logic](#) for more information about counters in shared allowances.

Parameters

Each allowance includes its own parameters.

Sharing Allowances Between Provider Contracts within a Subscriber Account

The following rules must be taken into account when sharing allowances:

- An allowance is linked to a provider contract, which can be itself linked to a parent provider contract
- A provider contract can access the allowances of its parent provider contract, but a parent provider contract cannot access the allowances of other provider contracts

Sharing Allowances between Provider Contracts from separate Subscriber Accounts

From a charge plan or a refill plan that is referenced by a provider contract (called “owner”), the user creates allowances that can be used from provider contracts that belong to other subscriber accounts. To share an allowance, the provider contract that owns the allowance specifies a unique share identifier at the creation of the shared allowance. The other provider contracts (called “members”) use this share identifier to reference the shared allowance. For more information about the Create Allowance Operator component, refer to the [SAP Convergent Charging Logic Components Reference](#) documentation.



NOTES

- The allowance plan of a shared allowance must be shareable
- In the Create Allowance Operator component, the user can decide if an allowance is shared or not

To build up a group of shared allowances, one or several provider contracts (owners) can create several shared allowances under the same share identifier. To use these shared allowances, the other provider contracts (members) must send allowance events through an Allowance Event Sender Operator component that is configured to use this share identifier. Therefore, only the Event-Based Trigger components of these shared allowances are activated and their logic are executed.



NOTE

A provider contract (member) can send an allowance event to a shared allowance only through an allowance that is not shared and that belongs to this provider contract. This rule is implemented in an allowance logic object that is not shareable and contains an Allowance Event Sender Operator component that is configured to send allowance events to shared allowances. For more information about this component, refer to the [SAP Convergent Charging Logic Components Reference](#) documentation.

BART Server Master Data

CDR (Consumption Detail Record)

A Consumption Detail Record corresponds to a computer record produced by a telecommunication system containing details of a service usage. These records:

- Represent a superset of the chargeable item related to the incoming record
- Are managed by the BART Server, from acquisition to rating steps

- Are stored in the CDR table of the BART database, which contains all the fields and properties of the incoming record

 **NOTE**

The “Consumption” term represents the generalization of the “Call” term which is more restrictive as only used in case of phone exchanges.

CDR fields

The CDR table of the BART database contains the following fields:

Property	Description
OID	Unique identifier of the CDR
BGRP_OID	Unique batch group identifier, which is used during the following processes: <ul style="list-style-type: none"> ■ Chargeable Items Consolidation ■ Chargeable Items Charging (batch execution mode)
NAME	Name of the CDR which must refer to a chargeable item defined in the Core database
USER_ID	The association of these fields refer to an access defined in the Core Server
SERVICE_ID	
SUBSCRIPTION_ID	Identifier of a subscription defined in the Core database and used during the following processes: <ul style="list-style-type: none"> ■ Chargeable Items Consolidation ■ Chargeable Items Charging (batch execution mode)
PARTITION_ID	Identifier of the database partition in which this CDR is stored
CHARGE_ID	Identifier of a charge component defined in the Core Server
CONSUMPTION_DATE	The consumption date of this CDR
STATUS	The status of this CDR, which can be: <ul style="list-style-type: none"> ■ NEW: The CDR is newly acquired ■ RATED: The CDR is successfully charged ■ ERROR: An error occurred during the CDR charging operation ■ DUPLICATED: The CDR is a duplicate of another existing CDR ■ IGNORED: The CDR has been ignored during a charging session ■ TO BE RERATED: The CDR is selected to be rerated for the next subscription ■ NO PROVISIONING: The CDR must be consolidated again before being charged. The CDR has already been consolidated but the consolidation failed because the CDR does not contain any subscription identifier ■ ONLINE SUBSCRIPTION: The CDR has been consolidated but is not valid because it refers to an online subscription
ACQUISITION_DATE	Date of the CDR acquisition process execution
ACQUISITION_ID	Identifier of the acquisition session which relates to this CDR
RATING_DATE	Date when this CDR has been sent to the Core Server for charging purposes
RATING_ID	Identifier of the acquisition session which relates to this CDR

ERROR_CODE	Error code returned by the Core Server during a charging operation
ERROR_DESCRIPTION	Explicit message related to the above error code
SOURCE	The source of the CDR, provided during the Chargeable Items Acquisition process
MAGIC_NUMBER	Property encoded in MD5 format and used to check the CDR uniqueness in the database
PROPERTIES	TV formatted string containing the properties of the related CDR.
SNAPSHOT_ID	Identifier of the counter snapshot related to the last subscription rerating operation

CDR properties

A CDR contains several properties stored in the `PROPERTIES` column of the CDR table using a TV format. Each property of a CDR is made up with the following attributes:

- Name, which corresponds to the name of the property
- Type, which corresponds to the type of the property's value (string, number or date)
- Value

The `PROPERTIES` column of the CDR table has a limited size (set by default to 4000 characters) which thus limits the number of concatenated properties and values. To increase this number, a `PROPERTIES` table has been implemented to store all existing properties. This table thus represents a CDR properties' dictionary which gives the possibility to use properties' identifiers instead of real full names:

Property	Description
OID	Unique identifier of the CDR's property, used as the tag part of a TV string
NAME	Name of the CDR's property
TYPE	Type of the CDR's property: <ul style="list-style-type: none"> ■ 0 for string values ■ 1 for number values ■ 2 for date values



EXAMPLE

Considering the following CDR:

```
1;JOHNSON;2;EDISON;3;777-554-2221;4;2008-01-03 11:55:26
```

The following values can be extracted:

- JOHNSON, value of property 1
- EDISON, value of property 2
- 777-554-2221, value of property 3
- 2008-01-03 11:55:26, value of property 4

Diameter Server Master Data

CCR (Credit Control Request)

Credit-control requests are sent by a Diameter client to request credit authorization for a given service. CCRs have the following format:

```
<Credit-Control-Request> ::=
  < Diameter Header: 272, REQ, PXY >
    < Session-Id >
      { Origin-Host }
      { Origin-Realm }
      { Destination-Realm }
      { Auth-Application-Id }
      { Service-Context-Id }
      { CC-Request-Type }
      { CC-Request-Number }
      [ Destination-Host ]
      [ User-Name ]
      [ Origin-State-Id ]
      [ Event-Timestamp ]
      *[ Subscription-Id ]
      [ Service-Identifier ]
      [ Termination-Cause ]
      [ Requested-Service-Unit ]
      [ Requested-Action ]
      *[ Used-Service-Unit ]
      [ Multiple-Services-Indicator ]
      *[ Multiple-Services-Credit-Control ]
      *[ Service-Parameter-Info ]
      [ User-Equipment-Info ]
      *[ Proxy-Info ]
      *[ Route-Record ]
      [ Service-Information ]
      *[ AVP ]
```

With the following substructures:

```
Multiple-Services-Credit-Control ::= < AVP Header: 456 >
  [ Requested-Service-Unit ]
  *[ Used-Service-Unit ]
  *[ Service-Identifier ]
  [ Rating-Group ]
  *[ AVP ]
```

CCR format



NOTES

- The "*" sign refers to multiplicity
- Highlighted AVPs in the above sample codes correspond to AVPs which are partially or not supported, whereas a highlighted "*" sign only means that multiplicity is not supported:
 - The Requested-Action AVP does not support refund actions
 - Multiple occurrences of the Used-Service-Unit AVP are not supported because the Core Server is only able to handle one property to inverse during reservation operations
- When Multiple-Services-Credit-Control is used in the CCR instead of Requested-Service-Unit and Used-Service-Unit, its AVPs are moved to the CCR command level

CCA (Credit Control Answer)

Credit-control answers are sent by the Diameter Server to a client who previously requested credit authorization for a given service. CCAs have the following format:

```
<Credit-Control-Answer> ::=
  < Diameter Header: 272, PXY >
    < Session-Id >
      { Result-Code }
      { Origin-Host }
      { Origin-Realm }
      { Auth-Application-Id }
      { CC-Request-Type }
      { CC-Request-Number }
      [ CC-Session-Failover ]
      [ Granted-Service-Unit ]
    * [ Multiple-Services-Credit-Control ]
      [ Cost-Information ]
      [ Final-Unit-Indication ]
      [ Credit-Control-Failure-Handling ]
      [ Direct-Debiting-Failure-Handling ]
      [ Validity-Time ]
    * [ Redirect-Host ]
      [ Redirect-Host-Usage ]
      [ Redirect-Max-Cache-Time ]
    * [ Proxy-Info ]
    * [ Route-Record ]
    * [ Failed-AVP ]
      [ Service-Information ]
    * [ AVP ]
```

With the following substructures:

```
Cost-Information ::= < AVP Header: 423 >
  { Unit-Value }
  { Currency-Code }
  [ Cost-Unit ]

Multiple-Services-Credit-Control ::= < AVP Header: 456 >
  [ Granted-Service-Unit ]
  * [ Service-Identifier ]
  [ Rating-Group ]
  * [ G-S-U-Pool-Reference ]
  [ Validity-Time ]
  [ Result-Code ]
  [ Final-Unit-Indication ]
  * [ AVP ]

Granted-Service-Unit ::= < AVP Header: 431 >
  [ Tariff-Time-Change ]
  [ CC-Time ]
  [ CC-Total-Octets ]
  [ CC-Input-Octets ]
  [ CC-Output-Octets ]
  [ CC-Service-Specific-Units ]
  * [ AVP ]

Final-Unit-Indication ::= < AVP Header: 430 >
  { Final-Unit-Action }
  * [ Restriction-Filter-Rule ]
  * [ Filter-Id ]
  [ Redirect-Server ]
```















NOTES

- The “*” sign refers to multiplicity
- Highlighted AVPs in the above sample codes correspond to AVPs which are partially or not supported, whereas a highlighted “*” sign only means that multiplicity is not supported:
 - The `Cost-Information` AVP is not provided in case of free charging because the Core Server does not return any amount when a free price plan component is used (a flat price plan component with a 0 amount must be used in such a situation)
 - The `Cost-Unit` AVP is not supported because the Core Server does not store any human readable description of a price plan
 - The `Credit-Control-Failure-Handling` and `Direct-Debiting-Failure-Handling` AVPs are not supported because they have to be configured on the SCP side
 - The `Final-Unit-Indication` AVP is partially supported because the Core Server does not support redirection or access restriction
 - The `Final-Unit-Action` AVP only supports the TERMINATE action
- When `Multiple-Services-Credit-Control` is not used in the CCR, AVPs of `Granted-Service-Unit`, `Final-Unit-Indication` and `Validity-Time` are moved to the CCA command level

7 Processes and Functions

The following table provides a list of dependencies between the available processes and the concerned technical components:

Process	Core Server	BART Server	Diameter Server	CRM System	ERP System
 Chargeable Items Acquisition	■	■			
 Chargeable Items Consolidation		■			
 Activation	■				
 Product Modeling	■				■
 Order Management	■			■	
 Data Files Bulk Loading	■				
 Guiding	■				
 Prepaid Accounts Refilling	■				
 Chargeable Items Rerating	■	■			■
 Tax Calculation	■				
 Chargeable Items Charging			□		
 Credit Control Messages Conversion			■		

- Mandatory
- Mandatory in an SAP ERP/FI-CA integrated landscape
- Optional

Activation Process

Process Description

The Activation process consists in generating internal events to trigger the recurring and one-shot events corresponding to a given date. The generation of these events depends on the next activation date. These generated internal events do not contain any property except the date which is used as a reference date to decide if a recurring or one-shot event must be computed or not to update the counters in contracts and subscriptions. For each event, a new charging or refilling operation is triggered.



NOTES

- According to business requirements, it can be necessary to perform activation operations in the future. To limit the number of activation operations and thus avoid performance latencies, it is possible to define a period of time during which activation operations relating to a given incoming request can be performed. If an incoming request leads to activation operations that are posterior to the defined period, the system refuses the incoming request. For further information, consult the description of the "PERIOD_LIMIT_ACTIVATION" parameter in the [SAP Convergent Charging Parameters Reference](#) documentation
- The next activation date is computed for each activation charge in contracts. It is the closest date between the date of the next recurring event and the end date for this activation charge. The end date is the closest end

date among the objects related to the activation charge (contract item, charge plan including parameters and account assignments, charge plan item, charge component).

For one-shot events, different subtypes of internal events can be generated by the system:

One-shot event subtype	Signification		
	Provider contract charging	Subscription charging	Allowance
CREATION	Not applicable	Not applicable	The event date of the allowance creation has been reached
ACTIVATION	The start date of the first period of an element of the provider contract has been reached	The effective date of the subscription has been reached	The Validity Start Date of the allowance has been reached
EXPIRATION	Not applicable	Not applicable	The Validity End Date of the allowance has been reached
SUSPENSION	An end date of a valid period of an element of the provider contract has been reached	The suspension date of the subscription has been reached	Not applicable
RESUMPTION	A start date of a valid period of an element of the provider contract has been reached	The resumption date of the subscription has been reached	Not applicable
CANCELLATION	Not applicable	The commitment date of the subscription has been reached and the subscription is cancelled	Not applicable
EARLY_CANCELLATION	Not applicable	The subscription has been cancelled before its commitment date	Not applicable

It is possible to design price plans to model one-off fees according to these types of events.



NOTES

- When an element is added to a provider contract, the related one-shot fees are not immediately computed by default. They are only computed when the customer service is used for the very first time or when a recurring fee is applied
- When an end customer creates or cancels a subscription, the related one-shot fees are not immediately computed by default. They are only computed when the customer service is used for the very first time or when a recurring fee is applied
- It is possible to force these computations by triggering the Activation process from the Admin+ user interface or another frontend external system

For recurring fees, only the reference date is used to apply or not the fee. The recurring fees can also be applied in a prorated mode according to the life cycle of the subscription or the provider contract. This is done to reimburse the period where service no use has occurred, and/or charge only the used period.

Activation and Allowances

When a provider contract is activated, all its allowances are also activated. The system first processes the allowances.

Activation and Linked Contracts

When the activation relates to a provider contract linked to a parent contract, both its recurring and one-shot events and the recurring and one-shot events of its parent contract are triggered. The system processes first the parent contract.

External Triggering of the Activation Process

The Activation process can be manually or automatically triggered:

- **Manual triggering:** To ensure that all recurring and one-shot fees have been triggered on a given subscription or provider contract at a given date (for example before invoicing a customer), the SAP power users can specify a code and a reference date to use
- **API triggering:** SAP CC provides some technical interfaces (Web Services, HCI) which give the possibility to trigger the Activation process. According to your business requirement, you can implement a manual triggering, a periodic triggering or another triggering mechanism. For further information about these technical interfaces, refer to their technical documentation:
 - Web Services: Consider the *Business Process Management* process component and included service operations relating the Activation operations
 - HCI: Consider the Active operations available in the `admin.hci` and `cmd.message` Java packages

Internal Triggering of the Activation Process

The Activation process can be directly triggered by SAP CC:

- **Automatic triggering:** Each time a chargeable item is received, the Chargeable Items Charging process automatically:
 - Activates the current charge activation using the consumption date of the chargeable item as the reference date
 - Rates this incoming chargeable item

NOTE

The chronology between recurring and one-shot fees on the one hand and usage fees on the other hand is always respected in order to ensure coherent counters values during chargeable item charging operations. If a late chargeable item is charged after the automatic triggering of the Activation process, the counters values may be different than expected.

- **Periodic triggering:** SAP CC provides a default scheduler which gives the possibility for the Updater to trigger the Activation process. This default scheduler is not synchronized with events of an external system, such as a billing cycle. As a consequence, if you want to synchronize the Activation process with your billing cycle, you must use the provided APIs to implement your own scheduler, as described above.
For further information about the configuration of the scheduler provided by default, consult the description of

the “ACTIVATION_SCHEDULER_ENABLED” parameter in the [SAP Convergent Charging Parameters Reference documentation](#)

Bulk Execution of the Activation Process

The Activation process can be performed in a bulk mode through a scheduler which wakes up at a given reference date. When a bulk Activation operation is performed, all recurring and one-shot fees from the previous action to this reference date are activated.

NOTE

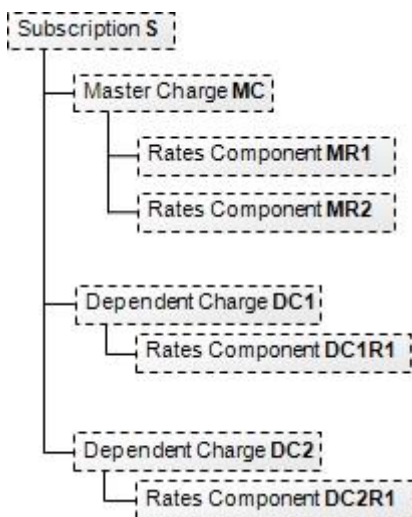
The Updater uses multiple parameters (such as a recurrence). For further information about these parameters, refer to the “Updater” section of the [SAP Convergent Charging Parameters Reference documentation](#).

Activation Process and Dependent Charges

If a master charge activated in a subscription or in a provider contract includes some recurring and/or one-shot rates in its price plans, the Activation process is automatically propagated to its dependent activated charges.

EXAMPLE

Considering a subscription S with a master charge MC which contains 2 recurring “Rates” price plan components MR1 and MR2. Considering that MC is combined to dependent charges DC1 and DC2 which themselves contain some recurring “Rates” price plan components DC1R1 and DC2R1:



When MR1 and MR2 are activated with a reference date D, the following propagation to all dependent charges occurs:

- 1- **MC**: MR1 is activated. If D implies to trigger the recurring fee, an amount is calculated and stored in the rating context. Otherwise, no propagation is performed
- 2- **DC1**: The dependent charge receives the rating context and DC1R1 recurring fee is applied (D is not taken into account to trigger DC1R1 – however, D can be used and tested into DC1R1). A transaction is generated if DC1 is concerned.
- 3- **DC2**: The dependent charge receives the rating context and DC2R1 is applied in the same way than for DC1. A transaction is generated if DC2 is concerned.
- 4- **MC**: MR2 is activated. If D implies to trigger the recurring fee, an amount is calculated and stored in the rating context. Otherwise, no propagation is performed
- 5- **DC1**: The dependent charge receives again the rating context and DC1R1 recurring fee is applied again (D is not taken into account to trigger DC1R1 – however, D can be used and tested into DC1R1). A transaction is generated if DC1 is concerned.
- 6- **DC2**: The dependent charge receives again the rating

context and DC2R1 is applied in same way than for DC1.
A transaction is generated if DC2 is concerned.

7- All generated transactions are returned

 **NOTE**

The number of generated transactions depends on the merging policy. If the **merge mode** is activated, a single transaction is generated, with an amount adjusted after each step.

 **NOTE**

Only recurring and/or one-shot rates price plan components belonging to dependent charges can be addressed.

Transactions Generated by the Activation Process

The transactions generated by the Activation process follow the same life cycle than the transactions generated during the execution of the [Chargeable Items Charging](#) process. These transactions are physically recorded into several files during the [Data Files Generation](#) function. The Bulkloaders then load the content of these files into the adequate database.

 **NOTE**

If a generated transaction corresponds to a recurring or one-shot fee related to a prepaid account with an insufficient balance, this balance is forced and thus becomes negative.

Product Modeling Process

Process Description

For functional information about this feature, refer to the description of the [Product Modeling](#) feature.

The Product Modeling process is a vendor process which consists in defining, building and managing charges, charge plans and offers with their associated business objects. This process includes the maintenance of this data in the system.

Different main master data objects are necessary according to the business services (online or offline charging, online rating) activated in SAP CC:

Business Service	Stateful Charging (online/offline) based on Provider contracts	Stateful Charging (online/offline) based on Subscriptions	Online Rating (stateless)
Main Master Data Objects			
Charges (Master)	■	■	■
Charges (Dependent)	■	■	

Charge Plans	■		
Offers		■	
Other Business Objects			
Pricing macros	■	■	■
Translation tables		■	■
Tier tables		■	■
Mapping tables and classes	■		■
Price tables and class	■		

This process is performed by the Updater, but is mainly implemented through the Core Tool user interface which:

- Eases the management of catalogs' life cycle
- Gives vendors the possibility to:
 - Define the (optional) relationships between his partners and himself (through dependent charges)
 - Assemble previous items in order to define the offers and the suboffer(s) which can be proposed to customers

The Updater is responsible for validating the catalog data and for managing their persistent storage in back-end database. It ensures the global consistency of all the master data objects used in the system.



NOTES

- In case of charging operations related to provider contracts, the Product Modeling process includes a step for assembling the charge plans in commercial products or offers. This step is performed by an external CRM or provisioning system. In the SAP Business Suite, SAP CC is directly interfaced with the SAP CRM system and an SAP user can assign some charge plans to a product or product bundle
- In case of charging operations related to subscriptions, the Catalog Building process may include a step for assembling the charges in offers outside from SAP CC. This operation is usually performed through the Core Tool user interface, but can also be performed by an external CRM or provisioning system interconnected with SAP CC using some provided Java and XML APIs (see the `CreateOfferOp` operation API)

Data Distribution

In the SAP Business Suite, the SAP CRM system retrieves the released charge plans and refill plans from SAP CC.

Cached Structures Refreshing

The Raters use catalog data objects for the rating and charging functions. Their data cached structures are periodically refreshed in order to have synchronized data. For further information about the refreshing possibilities, consult the "REFRESH_SCHEDULER_ENABLED" parameter description in the [SAP Convergent Charging documentation](#).



NOTES

- For test or implementation purposes, it is possible to force the resynchronization of the Raters directly from the Core Tool or Admin+ user interfaces. For further information about this feature, consult the "Refreshing rating instances" topic of the [SAP Convergent Charging Core Tool Online Help documentation](#)

- If the creation of the offers is managed in an external provisioning system, you can implement the `RefreshRatingInstancesOp` operation API

Process Limitations

- Some APIs are provided to create, modify, search and delete offers and associated business objects. The provided modification APIs are not “incremental” APIs, which means that users must specify the complete modified model instead of specifying only the modification to apply
- Contrary to offer parameters, translation tables and price plans, offers themselves are not managed chronologically. Performed modifications are thus immediately available and taken into account by other processes dealing with modified offers

Stateless Mode

- The Product Modeling process can be executed in a stateless mode, which is similar to the stateful mode, except the fact that no dependent charge can be specified and addressed. Only master charges are taken into account (and thus charged)
- As the charging function is not performed by the system, it is not possible to design a price plan with logic components (like operators or comparators) that use some properties based on information related to prepaid accounts (balance, empty limit threshold)

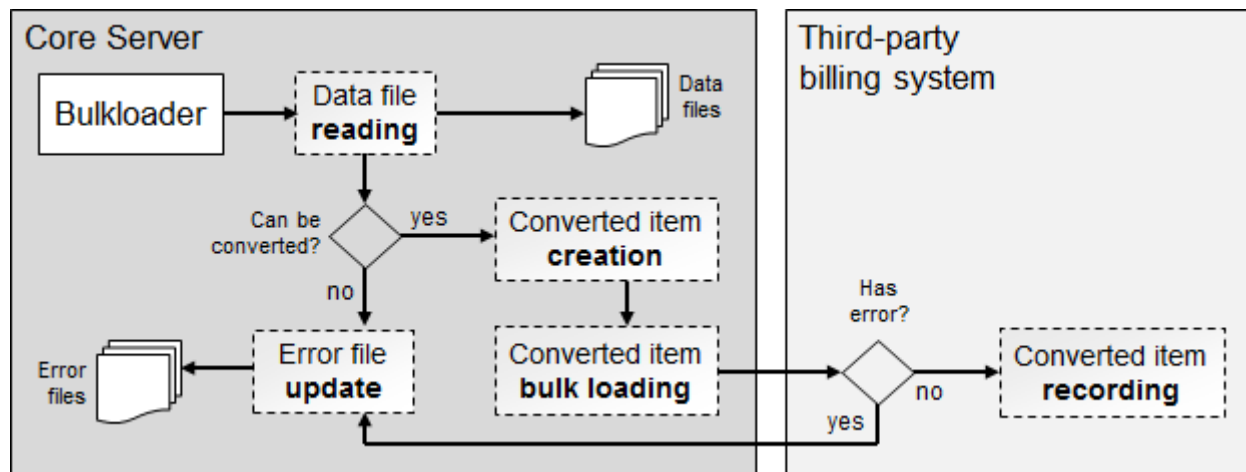
Data Files Bulk Loading Process

Process Overview

For functional information about this feature, refer to the description of the [Data Files Bulk Loading](#) feature.

The Data Files Bulk Loading process is used to load in a bulk mode the content of different data files into a third-party billing system. Bulk loading operations are performed by Bulkloaders, which work in conjunction with Raters in charge of generating these data files. The default behavior of the Bulkloaders consists in:

- Analyzing the content of the data files generated by the [Chargeable Items Charging](#) process
- Converting each element of each data file into an adequate item expected by the third-party billing system (such as billable items or consumption items when working in conjunction with SAP CI)
- Sending these converted items to the third-party billing system using the adequate communication channel (such as RFC-TCP/IP communication destinations when working in conjunction with SAP CI)



NOTES

- The default configuration of the Data Files Bulk Loading process:
 - Is dedicated to the use of SAP CI as a third-party billing system
 - Defines 2 reading channels that can be used by the Bulkloaders to handle data files containing charged items. As a consequence, the Data Files Bulk Loading process can be considered as a parallelized process able to simultaneously handle multiple data files
- Every Bulkloader handles the data files generated by a Rater which has the same instance number. For further information about instance numbers, refer to the [SAP Convergent Charging Installation Guide](#) documentation. For further information about the RFC-TCP/IP communication destinations used for bulk loading purposes, refer to the [SAP Convergent Charging Security Guide](#) documentation

Lock/Unlock Policy

The Data Files Bulk Loading process manages objects (directories and folders) shared with the [Data Files Generation of the Chargeable Items Charging process](#). To manage these concurrent accesses, a lock/unlock policy has been implemented. This policy relies on the data files' states and their associated filename extension:

- ".zip", ".gz" or ".csv" for files currently in use by Raters for writing purposes
- ".arc.zip", ".arc.gz" or ".arc.csv" for released files which are thus available for bulk loading operations
- ".arc.zip.bXX", ".arc.gz.bXX" or ".arc.csv.bXX" for data files in process by a given bulkloader, where "XX" corresponds to the instance number of the bulkloader (explained as 2 digits)
- ".rem.zip", ".rem.gz" or ".rem.csv" for files which are considered as archived and thus excluded from bulk loading operations. This additional "rem" extension is set when the `removeFile` attribute of the Bulkloaders' configuration is set to `false`

Running Bulkloaders continually scan the repositories defined in the configuration files, and handle all available released files. Every handled charged items data file is renamed by the concerned Bulkloader to prevent concurrent accesses, which ensures that its content can then be safely taken into account for bulk loading purposes.

Bulk Loading Mechanism

Every time they handle a data file, the Bulkloaders perform the following operations:

- The file is renamed to avoid concurrent accesses

- The header of the file is analyzed to retrieve the following information:
 - The version of the data file
 - The class name, which is used to determine the structure of the contained data
 - The line numbering flag which indicates whether the file is generated with line numbering or not
- The class name is analyzed to:
 - Verify the existence of this class in order to avoid loading obsolete data in the third-party billing system
 - Check the adequacy with the header of the file, and thus verify that no information is missing before loading data in the third-party billing system

Once these prerequisites have been checked, the Bulkloader reads every line of the data file, which corresponds to properties of an element (charged item, refill record or chargeable item) which must be:

- Converted using an adequate defined mapping
- Bulk loaded to the third-party billing system as a converted element

To ensure that all the elements of a data file are taken into account, the Bulkloaders re-use the control files generated during the [Data Files Generation](#) function. Due to the naming convention, every control file is associated to a configured reading channel which is thus able to handle its content. Control files contain the following information:

- The encoded identifier of the last item correctly written into the related data file. This identifier is a 4 bytes Long value which is updated by the related writing channel after each successful writing operation
- The encoded identifier of the last data file's row correctly sent to the third-party billing system. This identifier is a 4 bytes Long value which is updated by a reading channel after each successful loading operation
- Error information (Java error class name and exception message), when some errors occur during reading operations

When any error occurs:

- The erroneous data file is renamed with the suffixed ".err" extension
- Detailed information about the error is written into the control file for troubleshooting purposes

When a data file has been successfully and entirely loaded into the third-party system, the control file is automatically deleted by the Bulkloader.

In addition to control files, dedicated error files are also created by the Bulkloaders to:

- Provide detailed information about the errors which can occur during the conversion of the source items, can fail for the following main reasons:
 - The charged item mapping is not correctly configured (for instance, an item field is initialized with a wrong property)
 - The billable item mapping is not well configured (for instance, the item field X is mapped onto the billable item field Y but X and Y do not have the same semantics)
 - The charge which leads to the generation of the item does not provide any value for mandatory fields or provides invalid values (for instance, negative number for strictly positive number type field)
- Provide detailed information about the errors which can occur when the converted items are sent to the third-party billing system. In an integrated scenario with the SAP ERP/FI-CA component of the SAP Business Suite, another validation is performed by SAP CI, which can also fail due to specific constraints defined on SAP CI side
- Give administrators the possibility to perform related corrections, which is really important as such items represent effective consumptions which:
 - Have been authorized for customers
 - Must be as soon as possible loaded into the third-party billing system for money recovery purposes

The Bulkloaders can write error files using 2 different formats:

- Version 1, which only contains the original version of the rejected items
- Version 2, which contains the original version of the rejected items, and some messages describing the reasons why these items have been rejected



NOTES

- Version 2 is enabled by default for new fresh installations. During an upgrade of a SAP Convergent Charging system, this version is not updated, to ensure backward compatibility and preserve the potential already running integration with external systems dealing with error files
- It is recommended to upgrade the version of the error files, except if these files need to be managed by an external system which does not support comments within CSV files. To modify the version of the error files, refer to the [SAP Convergent Charging Configuration Guide](#) documentation

Version 1 error files

The filenames of the version 1 error files are made up with the following information, each element separated by a underscore (_) sign:

- CIT class name
- Writing channel identifier (3 digits number)
- File rolling policy (3 characters string such as "HOU" for hourly rolling policy)
- Writing channel index (2 digits number)
- Timestamp using the yyyyMMddTHHmssSSS format (e.g. 20141101T123456789)

The version 1 error files contain:

- One and only one header block containing the names and types of the fields which are contained in the error file
- One or several items, written using a CSV format

To figure out the reason why the items have been considered as erroneous, the Bulkloader log files must be analyzed.

Version 2 error files

The filenames of the version 2 error files are made up with the following information, each element separated by a underscore (_) sign:

- If the "fileSequencId" attribute is empty or missing:
 - CIT class name
 - Writing channel identifier (3 digits number)
 - File rolling policy (3 characters string such as "HOU" for hourly rolling policy)
 - Writing channel index (2 digits number)
 - Timestamp using the yyyyMMddTHHmssSSS format (e.g. 20141101T123456789)
- If the "fileSequencId" attribute is not empty:
 - SAP system identifier (3 alphanumeric characters string)
 - CIT class name
 - Writing channel identifier (3 digits number)
 - File rolling policy (3 characters string such as "HOU" for hourly rolling policy)
 - Writing channel index (2 digits number)
 - Timestamp using the yyyyMMddTHHmssSSS format (e.g. 20141101T123456789)
 - Sequence number (6-digit number from 000000 to 999999)

The version 2 error files contain:

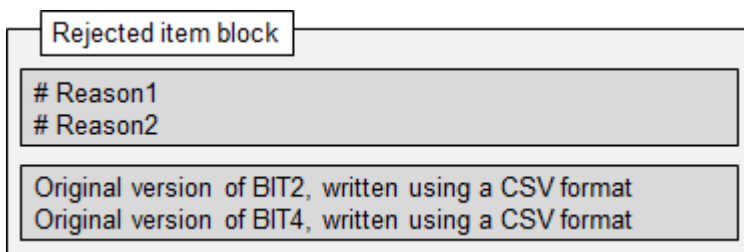
- One and only one header block containing:
 - The version of the error file
 - The name of the corresponding class
 - The names and types of the fields contained in the file
- One or several rejected item blocks, each block containing:
 - Messages informing about the reasons why the items have been considered as erroneous. These messages are stored in the file as comments (i.e. lines starting by the '#' character), and concern all the items written before the next block or before the end of the file
 - One or several items, written using a CSV format

 **NOTE**

In an integrated scenario with the SAP ERP/FI-CA component of the SAP Business Suite, the content of the version 2 error files depends on the context of the error:

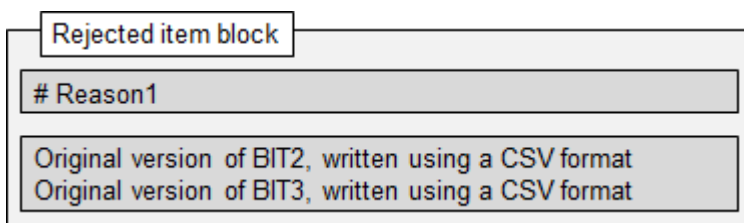
- When SAP CI rejects more than one converted item during a single function call, the rejected item block contains multiple items. When such a situation occurs:
 - The original versions of the rejected items are stored in the same block
 - The format of the written messages corresponds to the concatenation of the message id, message type (info: I or error: E), message number and the text of the message

Example: Considering 2 billable items BIT2 and BIT4, rejected by SAP CI due to reasons Reason1 and Reason2, then the error file will contain the following rejected item block:

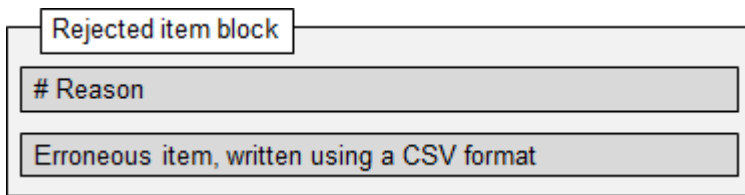


- When SAP CI rejects one converted item during a single function call because this item contains errors, the rejected item block contains all the items which relate to the same set of charged items (or refill records), even if they do not contain any error.

Example: Considering 3 billable items BIT2, BIT3 and BIT4, BIT2 rejected by SAP CI due to Reason1 and BIT3 relating to the same set of charged items (or refill records) as BIT2, then the error file will contain the following rejected item block:



- If the item validation is disabled on Raters side, erroneous items are stored in the data files, among the valid ones. These items are rejected by Bulkloaders during the conversion step. When such a situation occurs, the rejected item block only contains one message and one item:



Configuration Parameters

The Bulkloaders can be configured to fit specific needs. This configuration relies on 2 different kinds of parameters:

- Static parameters, whose name is entirely defined in the Java source code
- Dynamic parameters, whose name depends on a prefix specified in the Bulkloaders' configuration file named `cif.bulkloader.config.xml`. This prefix corresponds to the "adminParameterPrefixName" attribute of the "bulkReader" element.

EXAMPLE

Considering the following Bulkloader configuration file:

```
<bulkReader adminParameterPrefixName="POSTPAID_CIT_READER" bufferSize="1000" ... >
...
</bulkReader>
```

According to the content of the [Parameters Reference](#) documentation, the following dynamic parameters are automatically created:

- **POSTPAID_CIT_READER_ROOT_PATH**
- **POSTPAID_CIT_READER_CI_BUFFER_SIZE**
- **POSTPAID_CIT_READER_CHANNEL_COUNT**
- **POSTPAID_CIT_READER_REMOVE_FILE**
- And so on

Admin+ gives the possibility to list all the existing parameters, including static and dynamic parameters related to Bulkloaders. To adapt the behavior of a given Bulkloader to specific needs, some of these parameters can be hot-modified using Admin+.

NOTE

For further information about the configuration of Bulkloaders and the associated parameters, refer to the [SAP Convergent Charging Configuration Guide](#) and [SAP Convergent Charging Parameters Reference](#) documentations.

Order Management Process

Process Description

For functional information about this feature, refer to the description of the [Order Management](#) feature.

The Order Management process is made up with 3 distinct functions:

- A CSR business partner provisioning function, used to manage the subscriber accounts of customers (including their life cycle)
- A CSR commercial provisioning function, used to manage subscriptions or provider contracts (including their life cycle) associated to the customers
- An OSS/BSS technical provisioning function, used to manage the provisioning of accesses in order to connect technical resources located in the OSS to SAP CC

 **NOTE**

Due to the fact that the Order Management process modifies accesses information, it thus works in conjunction with the internal function named “guiding function”, in charge of maintaining an up-to-date cached structure containing accesses data.

Provisioning of Subscriber Account

The Subscriber Account Provisioning function is a CSR function. It consists in managing the customer’s subscriber accounts from an external CRM or provisioning system into SAP Convergent Charging. Integration between SAP CC and this system must thus be implemented.

 **NOTE**

In an integrated scenario with the SAP CRM, the Subscriber Account Provisioning function is periodically performed by the SAP CRM system which is responsible of the global consistency of all the master data.

Provisioning of Provider Contracts

The Provider Contracts Provisioning function is a CSR function. It consists in managing the customer’s provider contracts from an external CRM or provisioning system into SAP CC. Multiple actions can be performed on a provider contract during its life cycle:

- Creation
- Modification

When a customer (or business partner) subscribes to a vendor’s commercial product or offer, this process instantiates a provider contract by:

- Creating a provider contract in the CRM system
- Replicating the relevant data in SAP CC
- Validating the received data
- Creating the necessary activation charges
- Creating the necessary counters
- Creating and updating the necessary accesses (see the Access Provisioning function described below)

 **NOTE**

In an integrated *SAP Solution* scenario with the SAP CRM and the SAP ERP/FI-CA components of SAP Business Suite, this process is launched when a sale person has sold a product bundle to a business partner. He creates a provider order and a provider contract. If the activated product includes some assigned charge plans, the SAP CRM system will create the necessary charging data in SAP CC.

The Updater is responsible for validating the provider contract data and for managing their persistent storage in the database. It ensures the global consistency of all the master data objects used in the system.

The Provider Contract Provisioning function is made up with two different steps:

- A CRM or provisioning system creates or maintains a provider contract by using the Web Services of SAP CC
- The Core Server validates and completes this customer master data:
 - It validates the data
 - It assigns a batch rating group to the provider contract: If the contract is linked to a parent contract (to share a pool of counters), it has the same batch rating group; otherwise it performs the batch rating group assignment function that can be customized
 - It creates or maintains automatically the necessary charge activations for setting up the charging functions. It creates in the provider contract the relevant counters and launches eventually an activation process (see the Activation process)



NOTES

- An activation charge is created for each charge in the activated charge plan whatever the type of rates that are defined in the price plan of the charge
- The provider contracts and their contract items are not directly managed chronologically, but multiple versions can however successively exist for different periods of time
- A provider contract revision is used to manage the structure of a provider contract. Several contract item revisions are used to group together the contract item parameters, account assignments and user technical identifiers. As a consequence, if the contract revision or a contract item revision is modified at a given date, a new version of this revision is automatically created. According to a given date, the system is then able to retrieve the right provider contract revision and its contract item revisions which correspond finally to a unique version of the contract that has been ordered in the CRM or provisioning system

Provisioning of Subscriptions

The Subscription Provisioning function is a CSR process. It consists in managing the customer's subscriptions. Multiple actions can be performed on a subscription during its life cycle:

- Creation
- Modification
- Resumption
- Suspension
- Deletion

When a customer subscribes to a vendor's offer, this process instantiates a subscription by:

- Recording the parameters related to the offer
- Creating the customer balances and/or counters according to the chosen offer and associated options



NOTES

- It is possible to postpone the counter instantiation to the first future charging operation. To implement this mechanism, the subscription model (used as argument at creation time) can be provided without any counter. Adequate counters are then automatically instantiated according to overload rules specified in the offer. As this behavior can lead to a latency peak during the first charging operation, **we highly recommend to instantiate counters at the subscription creation time**

- Subscriptions are not directly managed chronologically, but multiple versions can however exist through the subscription contexts. A subscription context is an object which groups together the subscription parameters and the translation table exceptions. As a consequence, if the subscription context is modified at a given date, a new version of this context is automatically created. According to a given date, the system is then able to retrieve the right subscription context which corresponds to a unique version

Provisioning of Accesses

The Access Provisioning function gives the possibility to manage accesses which are used to interface SAP CC with an external OSS.

The Access Provisioning function is managed by:

- The external CRM or provisioning system if the access is dedicated to subscriptions
- The Core Server if the access is dedicated to provider contracts

Provisioning	Function Description
Accesses dedicated to subscriptions	<p>Giving the possibility to define and/or associate an access to a subscription in order to reach the adequate price plan and charging plan, the Access Provisioning process is:</p> <ul style="list-style-type: none"> ■ Mandatory if the offer contains some usage fees, as accesses represents the only way to associate an incoming chargeable item to the right subscription ■ Optional if the offer only contains recurring and one-shot fees, as the Activation process can be used to charge the consumed usage
Accesses dedicated to provider contracts	For each contract item, SAP CC creates an access for each User Technical Identifier (UTI) and each Service Identifier declared in the related activated charge plan or refill plan.

An SAP User can manually perform this process for test or training purposes by using the Core Tool user interface with accesses dedicated to a subscription.



NOTES

- A mediation system can be introduced between the OSS and the BSS when the OSS brings into play multiple systems
- SAP CC implements an optional mechanism which gives the possibility to manage accesses according to defined time periods (continuous or not), and thus assign different accesses on the same subscription for a given customer. When this chronology mechanism is unused, an unlimited time period is assigned to the access
- An access can refer to both subscriptions and provider contracts
- Accesses are also created for the contract items which activate a refill plan. The same mechanism is performed by SAP CC

Prepaid Accounts Refilling Process

Process Overview

For functional information about this feature, refer to the description of the [Prepaid Accounts Refilling](#) feature.

Every subscriber account contains a prepaid balance which is used for prepaid transactions and whose value can be decreased by the Usage Rating and Subscriber Charging process. The Prepaid Accounts Refilling process is a process which impacts the prepaid balances and/or counters by increasing their values, and which can also create and/or use allowances.

This process can be triggered by both external and internal events. It uses refill trees to compute complex refilling operations which can lead to the generation of data files named “refill records” and containing transactions.

External Events

External events are named “Refill Items” and contain information related to the way a prepaid account balance must be refilled. These external events can correspond to:

- Refill operations, which increase the balances of prepaid accounts
- Negative refill operations, which are used to:
 - Cancel a previous refill operation
 - Transfer an amount from one prepaid account to another
 - Decrease the balances of prepaid accounts



CAUTION

As described in the “Subscriber account” [master data section](#) of this guide, prepaid balances are connected to a multicast notification mechanism used to send alerts when a threshold has been reached.

Negative refills are considered as forced debits which can decrease a prepaid balance under its “Empty Limit” property, and thus:

- Trigger an amount alert if such an alert has been associated to the prepaid balance
- Automatically trigger a refill operation (if such a behavior has been defined in the refill logic) to refill the prepaid balance and avoid any service cut off

If an automatic refill operation fails for any reason (system failure, insufficient money on the bank account, and so on), the system cancels it using a negative refill operation. This negative refill operation decreases the prepaid balance, which thus goes back to a value which is:

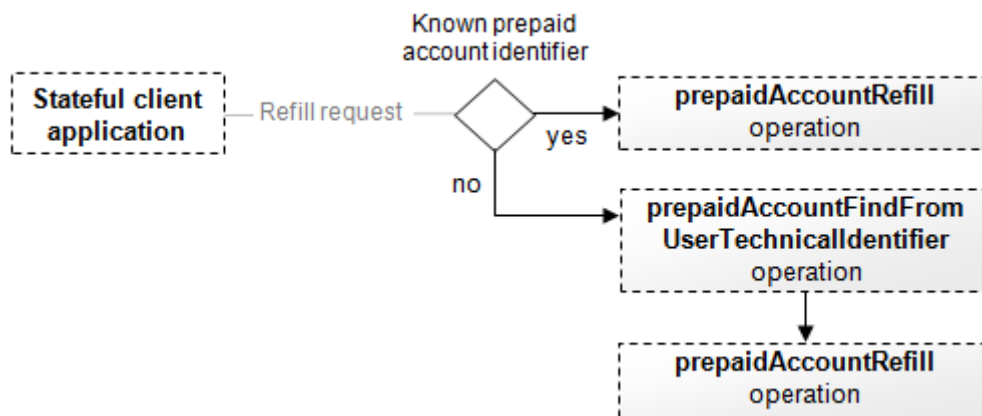
- Similar to the value before the automatic refill operation if no service has been consumed during this period of time
- Inferior to the value before the automatic refill operation if a service has been consumed during this period of time. Precautions should be taken to avoid such a situation, as it may lead to a loss of money

To avoid any infinite loop which would decrease the global performance of the solution, no automatic refill operation is triggered if the “Empty Limit” property of a prepaid balance is reached because of a negative refill. This ensures that amount alerts will not be sent multiple times.



NOTE

The identifier of the prepaid account which must be refilled must be provided to the “prepaidAccountRefill” operation of the “Refill Management” web service. When this identifier is not known, a previous call to the “prepaidAccountFindFromUserTechnicalIdentifier” operation must be performed in order to retrieve this prepaid account identifier.



Internal Events

Internal events correspond to recurring and one-shot refills which are auto-generated during the execution of the refilling process. This auto-generation can be due to:

- The activation process, which triggers the recurring refills at a given date
- A modification of the provider contract's state (such as its suspension, resumption, and so on), which triggers the related one-shot refills
- An event which occurs on a given account (such as a prepaid balance falling under a threshold)



EXAMPLES

- A recurring refill can be used to periodically refill the balance of a prepaid account
- A one-shot refill can be used to refill the balance of a prepaid account when the provider contract is activated (welcome refill)
- An account event refill can be used to automatically refill the balance of a prepaid account in case this balance falls under a given threshold (after a service consumption)

Process Execution

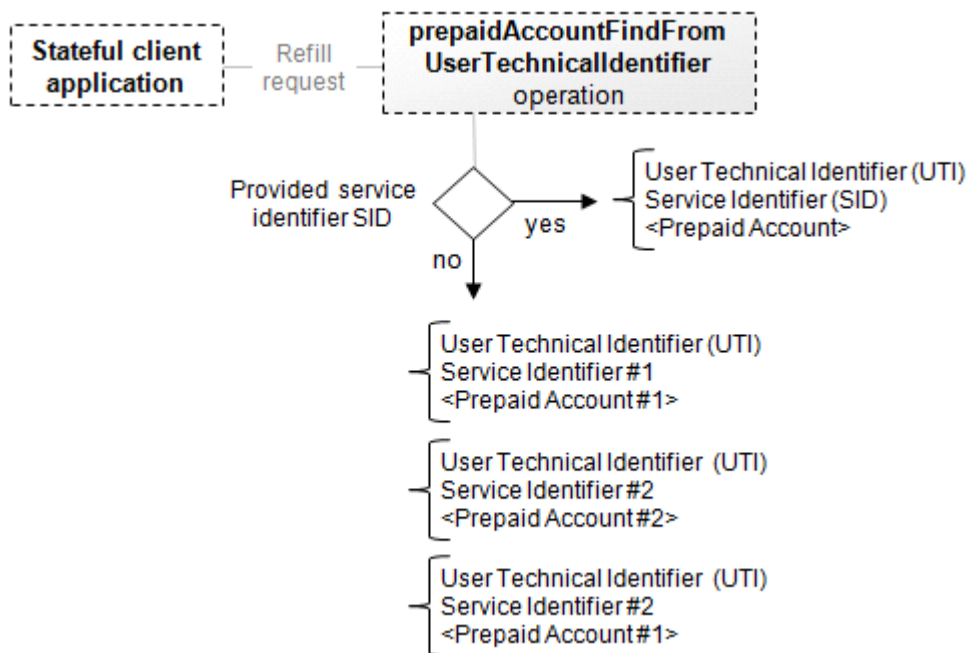
Refill requests are sent by client application to the Updater using the "Refill Management" web service. This web service contains the following operations:

- "PrepaidAccountRefill", which gives the possibility to refill the balance of a given prepaid account according to a provided refill item
- "PrepaidAccountFindFromUserTechnicalIdentifier", which gives the possibility to identify a prepaid account from a UTI and return the information which are necessary to refill its balance

The "PrepaidAccountFindFromUserTechnicalIdentifier" operation is used by client applications such as SAP CRM in order to retrieve the list of prepaid accounts related to a provided User Technical Identifier (optionally completed with a Service Identifier).



EXAMPLE



NOTES

- When specified together, the User Technical Identifier and Service Identifier couple represents an access which refers to a given prepaid account. Contrary to accesses created in the charging part of the solution, refill accesses are transparent for users and only used internally
- In an integrated landscape, it is highly recommended to configure the identifier of the service used to refill prepaid accounts. When specified in the configuration, client applications using the "PrepaidAccountFindFromUserTechnicalIdentifier" operation can thus immediately determine the prepaid account to use, whatever the number of returned items is.

The "PrepaidAccountRefill" operation is used by client applications such as the SAP CRM to perform refilling operations on prepaid accounts. This operation requires information such as:

- The identifier of the concerned subscriber account
- The identifier of the prepaid account whose balance must be refilled
- The amount of the refill
- The refill item, which contains all the properties of the external event

Every request received by the Updater leads to a guiding operation in order to determine the concerned Rater. The Updater then transfers the request to the Rater, which performs the following operations:

1. **Refill activation**, used to generate the recurring and/or one-shot events associated to the refill item. When internal events are generated, they are suggested to the refill logic in order to create the corresponding transactions
2. **Refill logic execution**, applied on the refill item in order to:
 - Generate the corresponding transaction
 - Create and use allowances when necessary
3. **Creation of allowances**, and execution of the related one-shot trigger if such a component exists in the allowance logic
4. **Data files generation**, which gives the possibility to:

- Record the generated transactions as "refill records" into data files and containing information such as:
 - The catalog data (such as the code of the applied refill plan)
 - The origin of the refill (one-time, periodic, voucher, and so on)
 - The properties of the refill item
 - The computed properties (such as the added credit per service)
 - The information related to the prepaid account (balance, state, and so on)
- Generate charged items data files according to the configuration of allowances

When the refilling operation is completed, the "PrepaidAccountRefill" operation returns the status of the execution to the client application, possibly associated to an error message in case of any failure.



NOTES

- For more information about the activation process, refer to the description of the [Activation process](#)
- For more information about the generated data files, refer to the [Data Files Generation function of the Chargeable Items Charging Process](#) section located afterwards

Chargeable Items Rerating Process

Process Overview

For functional information about this feature, refer to the description of the [Chargeable Items Rerating](#) feature.

A running SAP CC system can face following situations such as:

- Missing or erroneous incoming chargeable items
- Incorrect configuration of the Chargeable Items Charging process, which can lead to situations such as:
 - Incorrect access, subscription or provider contract
 - Errors in an offer, in a translation table, in a pricing macro or in a charge plan
 - And so on

When such problems occur, the Chargeable Items Rerating process gives the possibility to rate again a set of chargeable items related to postpaid accounts which have previously been charged and in order to:

- Correct potential errors and faults by reprocessing the original events
- Restore data consistency

This process:

- Operates from rerating data files, which are generated by the "Rerate File processor" (which may require some additional developments in order to fit specific needs of a third-party billing system) during the Data Files Generation function of the Chargeable Items Charging process
- Implies an adequate configuration of the data file processors used during the Data Files Generation function of the Chargeable Items Charging process, as the third-party billing system must be able to cancel billing data before executing operations
- Can be driven by either the BART Server component, or SAP CI

Process Limitations

- The Chargeable Items Rerating process should not be triggered if a billing cycle is running on the billing system side

- Rerating operations can only be performed on:
 - All provider contracts which are not concerned by online charging operations and do not refer to prepaid accounts
 - Offline subscriptions (due to incompatibility with the online and batch modes of the Chargeable Items Charging process)
- Rerating operations are not compliant with shared allowances in the following contexts:
 - If a charge plan or a refill plan has created (or can create) shared allowances. All operations related to the provider contracts that use this plan are rejected
 - If a plan has created (or can create) allowances that are configured to send events to shared allowances through an Allowance Event Sender Operator: All operations related to the provider contracts that use this plan are rejected

Execution Modes

Rerating driven by the BART Server

The Core Tool user interface gives users the possibility to trigger rerating operations which are executed by the BART Server system with the Core Server system in background.

The Chargeable Items Rerating process handles multiple subscriptions and provider contracts at the same time. As a consequence, this process is considered as a bulk process which performs the following operations:

- **Rerating session opening:** Performed by Updaters, this operation:
 - Checks the configuration of SAP CC to ensure that rerating is enabled
 - Checks that no Activation operations are running
 - Checks the availability of counter snapshots
 - Retrieves the number of subscriptions and provider contracts concerned by the rerating operation (for computation and information purposes)
 - Locks the system to avoid parallel concurrent rerating sessions
- **Subscription and provider contracts selection:** Performed by Updaters, this operation:
 - Builds the list of subscriptions and provider contracts concerned by the rerating operation. It includes consistent collections of provider contracts when a contract is linked to a parent contract. The parent contract and all its linked contracts belongs to a collection
 - Stores this list in the `RERATING_SESSION_ORDER` table of the BART database
 - Locks the charged item data files and synchronizes the Bulkloaders in order to bulk load all billable items which could be reversed in the further operations
- **Rerating session starting:** Initiated by Raters and performed by the BART Server system, this operation groups the list of subscriptions and provider contracts concerned by the rerating operation by batch rating groups and:
 - Loops on each batch rating group in order to restore every subscriptions or provider contracts, element per element and sorted by consumption date. This restoration operation is performed in multiple steps:
 1. The subscription or provider contract is locked
 2. All the counters' values of the locked element are restored according to saved snapshots
 3. The element's state is updated both in memory and in database
 4. A call is sent to SAP CI in SAP ERP/FI-CA to check the reversibility of the billable items associated to the subscription or provider contract for a given restoration date
 5. If SAP CI can reverse the billable items related to the element using the provided restoration date, they are asynchronously reversed. Otherwise, SAP CI is able to provide another valid restoration date for this element (named `fromDate`) which can be used during the next rerating operation performed on this element. If such a situation occurs, the rerating operation related to this subscription or provider

contract is cancelled. The element is then unlocked to authorize further rerating operations, and its state is updated accordingly.



NOTES

- The billable items reversibility check is an SAP CI operation, which is thus only available when SAP CC is integrated with it in an *SAP Solution* scenario. SAP CC uses the FKK_BIX_BIT_REVERSE_CHECK_CC JCo function.
- Billable items cannot be reversed if they have already been billed and if the configuration of SAP CI does not allow to reverse billable items which have already been billed
- When a billable item is considered as reversible, SAP CI:
 - Removes it if this billable item has not been previously billed
 - Creates a new billable item in the next billing cycle in order to compensate the original one if the original billable item has already been billed
- In a system landscape with several SAP ERP systems, SAP CC selects the relevant ERP system by using the customer management area information that is set in the subscriber account relating to the locked provider contract.

- Rerates all subscriptions or provider contracts of every batch rating group
- **Rerating session stopping:** Performed by the BART Server system, this operation:
 - Loops on each subscription or provider contract concerned by the rerating operation to unlock it
 - Unlocks the system to authorize further rerating sessions



NOTES

- The Chargeable Items Rerating process is only meaningful in an **offline** rating mode. A reference date is required in order to retrieve a list of CDRs to rerate.
- The Chargeable Items Rerating process can be interrupted at any time by power users. When such a situation occurs, every begun batch rating group is finished before stopping the running rerating operation.
- A dedicated command gives the possibility to retrieve the status of a running rerating operation. It gives the possibility to evaluate the advancement of the operation, with information such as the number of already rerated elements.

Counter Snapshots Management

The Counter Snapshots mechanism is based on consumption date provided to the Chargeable Items Charging process. This mechanism assumes that CDRs are rated subscription per subscription (or provider contract per provider contract) ordered by increasing consumption date. Each subscription or provider contract maintains its own snapshot state, but snapshot policy is defined globally for all subscriptions and provider contracts. The snapshot default policy is to backup counters each day at midnight, which implies that the first item whose consumption date is after midnight triggers the Counter Snapshots mechanism.



NOTES

- Daily counter snapshots are saved, whatever the consumption frequency. If a single consumption occurs during a whole month, all previous daily snapshots are recorded. This behavior gives the possibility to start a rerating operation to any anterior date
- The Activation process also triggers the Counter Snapshot mechanism



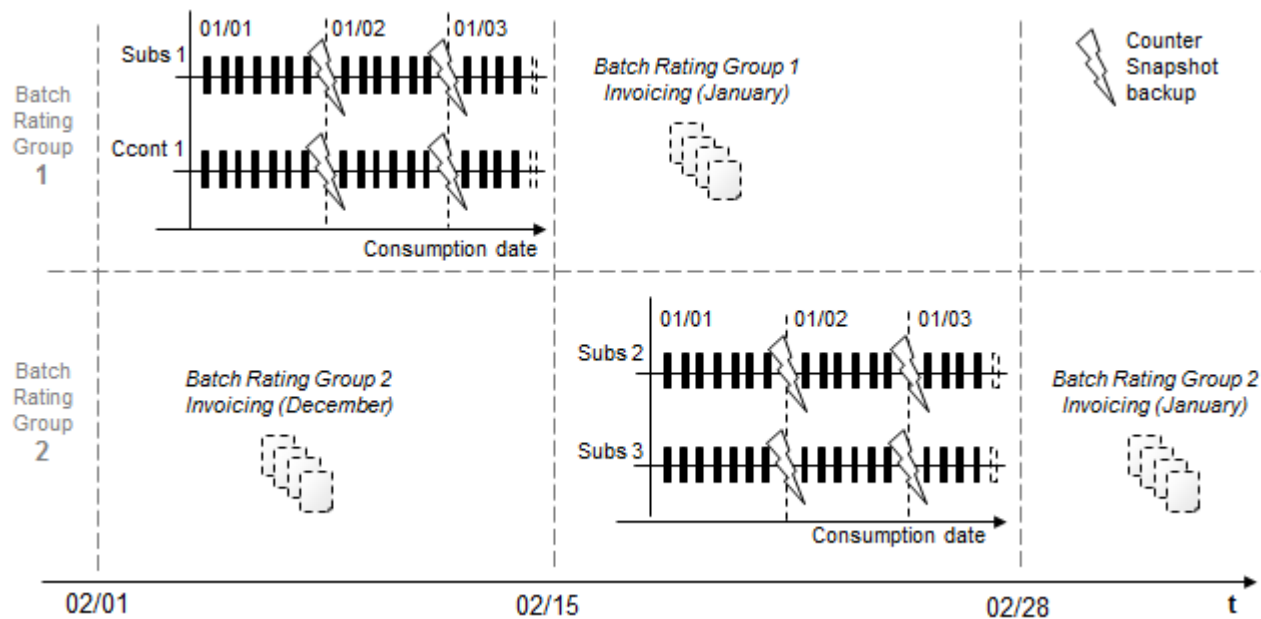
EXAMPLE

A third-party billing system performs an invoicing operation on a set of subscriptions and provider contracts grouped according to different batch rating groups in order to:

- Split these subscriptions and provider contracts in 2 different periods
- Balance the global Usage Rating operation

The rating of the entire previous month is done:

- From 01 to 15 of each month for the first group
- From 15 to 31 of each month for the second group



Explanation:

An error is detected in generated invoices generated between 02/15 and 02/28, due to a modification of an offer performed on 01/15. It is possible to rerate some subscriptions in both rating groups from the 01/15 snapshot, which limits the number of CDRs to rerate to a half month only. CDRs whose consumption date is greater than 01/15 are thus selected by the BART Server, even if they were rated after 01/02 (as the rating date is not taken into account for CDR selection).

To implement the Counter Snapshots mechanism, a counter named "next_snapshot_date" is added to each subscription and provider contract. This counter is used by the Activation process to trigger the Counter Snapshots mechanism (the Activation process is itself called by the Events Pricing operation before its execution).



NOTES

- If the event date overlaps several snapshot times and thus introduces latency, the Counter Snapshots mechanism may create multiple snapshots
- A COUNTER_SNAPSHOT table is used to store the different snapshots. The "COUNTER_SNAPSHOT_COUNT" parameter gives the possibility to configure the number of snapshots to store for each subscription and provider contract. For further information about this parameter, refer to the [SAP Convergent Charging Parameters Reference](#) documentation

Snapshots Backup

Counter snapshots are stored in the provisioning cache managed by Raters, using a proprietary format. These snapshots contain information such as:

- Last activation dates
- Next activation dates
- Next snapshot date
- Validity end dates of allowances
- Counters described in charge activations, provider contracts, provider contract items and allowances

Snapshots Restore

The Counter Restore operation is an operation which takes into account a given date (which does not necessary correspond to a snapshot backup date) to retrieve a list of counter snapshots which must be restored. Each retrieved snapshot is then analyzed in order to:

- Restore the adequate values of:
 - Counters related to subscriptions, provider contracts and allowances
 - Validity end dates of allowances
- Delete allowances created after the given date



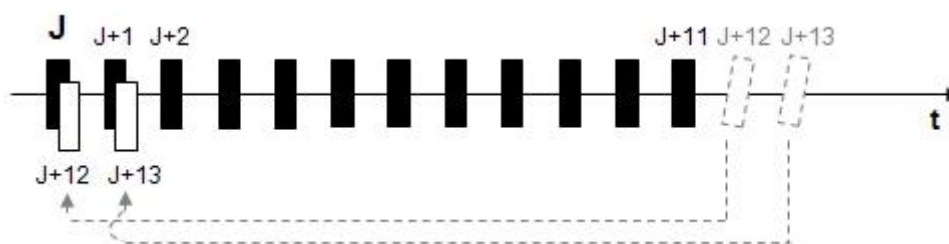
NOTE

As subscriptions and provider contracts content can evolve as time goes by, it is necessary to compare the snapshots with their associated subscriptions or provider contracts in order to delete counter(s) or create new one(s). For example, if a snapshot concerns a subscription which has 6 defined counters, but only 4 stored counter values, the 2 additional counters of the subscription must be deleted to match the snapshot counters number. If a snapshot concerns a subscription which has 6 defined counters, but 7 stored counter values, the additional counter of the snapshot is automatically created to match the number of counters associated to the subscription.

As allowances can evolve as time goes by, it is also necessary to compare the counter snapshots with the allowances related to their associated provider contracts, in order to restore the validity end dates of allowances or to delete allowances. For example:

- If a snapshot concerns a provider contract containing an allowance whose current validity end date is different from the one stored in the snapshot, the validity end date of the allowance must be modified to match the date stored in the counter snapshot
- If a snapshot concerns a provider contract containing an allowance which was not present when the snapshot was created, the allowance must be deleted to match the content of the snapshot.

As the snapshots storage capacity depends on the configuration, it is only possible to restore limited snapshot backup dates. If the reference date is anterior to this limit, Java exceptions are thrown to indicate that the subscription or provider contract cannot be restored. The example below explains that only 12 snapshots can be stored, and that rerating is thus only possible on 12 past days (J snapshot is replaced by J+12 one on day 13, and so on):



Subscriptions and provider contracts Lock / Unlock

All along the execution of the Chargeable Items Rerating process, locks are set and unset on handled elements such as subscriber account, subscriptions, transactions, and so on. These locks are set to avoid concurrent accesses from parallel operations such as charging operations.

When a parallel operation tries to lock an element which is already locked, a Java exception named `CannotLockObjectException` is thrown in order to indicate that:

- The subscription or provider contract is already locked
- No CDRs' state can be updated, except if a specific CDR error state is used to update all the CDRs of this charging session
- The charging session related to the subscription or provider contract must be stopped

In case of execution or server failure, a subscription or provider contract can be definitively locked, and thus require a manual operation to unlock it. Dedicated APIs named `unlockSubscriptionOp` and `UnlockChargingContractOp` are available to unlock a locked element (or a set of elements related to a given charging session) and thus gives the possibility to perform subsequent charging operations.



NOTE

When previous rerating operation(s) failed, the unlocked subscriptions or provider contracts have inconsistent states.

Process Limitations

- Modifying a counter of a subscription using the `modifySubscriptionOp` API is not recommended, because this modification is performed immediately, not recorded in counter snapshots, and thus not available for rerating operations
- On the one hand, batch charging operations store batch charging sessions and associated CDRs in the BART database before sending data to the third-party billing system. On the other hand, online charging operations lead to the creation of charged item data files which are **directly** loaded in the third-party billing system by Bulkloaders, and thus not stored in the BART database. Rerating operations require to cancel billable items which may come from both online and batch charging operations. As no incoming event related to an online charging operation is stored in the BART database, it is thus not possible to perform rerating operations on provider contracts concerned by online charging operations

Rerating driven by the SAP ERP/FI-CA Component of the SAP Business Suite

The Core Tool user interface gives users the possibility to initialize rerating operations which are executed by the SAP ERP/FI-CA component of the SAP Business Suite. Every rerating operation takes into account a list of provider contracts which must be rerated. When the operation is successfully completed, a message inform that the provider contracts have been sent to SAP ERP/FI-CA; otherwise, a message warns the SAP user that an error has occurred during the communication with the server system. For further information about the behavior of rerating operations, refer to the [SAP Convergent Charging Core Tool Online Help](#) documentation.

During rerating operations, SAP CC uses the following communication channels:

- XML over HTTP, to initialize the rerating process via the Core Tool user interface
- SOAP over HTTP, when receiving requests from SAP CI
- RFC over TCP/IP, to initialize a rerating session

**NOTE**

For further information about these communication channels, refer to the [SAP Convergent Charging Security Guide](#) documentation.

A rerating operation performed by the SAP ERP/FI-CA component of the SAP Business Suite is made up with the following steps (refer to the schema in the [Chargeable Items Rerating](#) section to get an overview of this execution):

1. Using the Core Tool user interface, a user initializes a rerating operation by creating a list of provider contracts which must be sent to SAP CI in relevant SAP ERP systems for rerating purpose
2. The Updater sends a rerating request to each involved SAP CI
3. Each SAP CI requests SAP CC to search for all the provider contracts which must be recharged together. If no error occurs, SAP CC returns a message containing the identifier of all the provider contracts related to the specified provider contract
4. SAP CI asks SAP CC to lock the collection of provider contracts in order to prevent any use of these provider contracts by other business processes
5. SAP CI requests SAP CC to prepare the asynchronous execution of the rerating process. The Raters release the in-use data files and the Bulkloaders process them. During this step, SAP CI regularly polls SAP CC which returns a message containing the status of the preparation (running, success, or in error)
6. SAP CI then requests SAP CC in order to find the restoration points of the provider contracts (state of each provider contract at the specified date). SAP CC returns a message containing the restoration points: identifier of the provider contract, range of charged transaction sets, and so on. The charged transaction sets gives SAP CI the possibility to determine the consumption items which must be recharged
7. SAP CI then requests SAP CC to restore the state of the collection of provider contracts at a specified date. The values of the different counters of these provider contracts are set to the values they had at that date. SAP CC returns a message containing the result of the restoration
8. SAP CI then requests SAP CC to recharge the set of chargeable items related to the collection of provider contracts. SAP CC returns a message containing the result of the recharging operation
9. SAP CI finally requests SAP CC to unlock the collection of provider contracts which has been previously locked for the recharging operation. SAP CC returns a message containing the result of this operation

**NOTE**

Refer to the [SAP Convergent Invoicing](#) documentation for more information about this process.

Chargeable Items Charging Process

Process Overview

For functional information about this feature, refer to the description of the [Online Charging](#), [Offline Charging](#), [Deferred Charging](#) and [Manual Capturing](#) features.

SAP Convergent Charging manages two different types of events:

- External events (chargeable items), which aggregate information corresponding to the way customers use provided services

- Internal events, which represent auto-generated events such as recurring events, contract events (creation, suspension, and so on), account events (threshold overtaking) or allowance events

The Chargeable Items Charging process is a process dedicated to the charging of these external and internal events. It handles these events within 5 main functions:

- **Rating contexts generation**, which gives the possibility to retrieve all the information which is necessary to charge both internal and external events
- **Events Pricing**, which gives the possibility to execute the defined price plans in order to calculate the different amounts (prices, taxes, and so on) related to the concerned events and allowances
- **Events Charging**, which gives the possibility to execute the defined charging plans in order to:
 - Retrieve the list of accounts with their associated balances
 - Charge these balances with the amounts calculated during execution of the previous function
- **Allowance Creation**, which gives the possibility to create allowances and execute the one-shot trigger components declared in the concerned allowance logic objects (and which relate to a creation event)
- **Data Files Generation**, which gives the possibility to create different data files containing all the information required by a third-party billing system for further actions such as billing operations, rerating operations, dispute management, and so on
- **Immediate Threshold Refilling**, which gives the possibility to trigger immediate refills potentially defined into refill plans and related to the concerned accounts

All over the execution of the Chargeable Items Charging process, the different events are technically materialized by transactions which contain information such as:

- Transactions details
- Tax information (retrieved from tax settings of both subscriber accounts and customized charges)
- Commands and associated arguments
- And so on

These transactions are enriched by the different functions and are successively considered as:

- **Rated** transactions, generated at the end of the price plans execution
- **Charged** transactions, created during the execution of the charging plans (for each account balance with must be charged) – A rated transaction can be split in several charged transactions
- **Charged and Taxed** transactions, which represent:
 - The Charged transactions enriched with the tax data calculated during the execution of both price plans and charging plans and applied on the price amounts computed during the execution of the price plans
 - The final data which are used by the data files generation function for persistency purposes

The Rated transactions can be returned in the response of the operation execution. For optimization and performance purposes, the content of these transactions can be filtered at Charge Plan Item level, using a Response Charged Item mapping.



NOTE

The details filtering technical mechanism is only available:

- When using the Online, Offline, Best-effort, Inverse, Blank, and Session-Based execution modes
- For operations relating to Provider Contracts, executed in stateful mode



CAUTION

The details filtering technical mechanism is not available for charging operations performed via Web Services.

Precision & Rounding Modes

During the execution of the process, the precision and rounding mode used during calculation operations can be configured to fit specific needs. This configuration concerns counters, transactions, transaction details and taxes, and can be configured differently for each type of these business objects.

The precision can be configured according to 2 different modes:

- A **manual** configuration which gives the possibility to specify a 0 to 6 digits precision (possible values are 0,1,2,3,4,5,6)
- An **automatic** configuration according to the currency precision (value: -1)

The rounding policy can be configured according to 3 different modes:

- **TRUNCATE** (Rounding mode to round towards zero), which never increments the digit prior to a discarded fraction (i.e., truncates). This rounding mode never increases the magnitude of the calculated value (e.g. 0.9 gives 0.0)
- **RAISE** (Rounding mode to round away from zero), which always increments the digit prior to a nonzero discarded fraction. This rounding mode never decreases the magnitude of the calculated value (e.g. 0.1 give 1.0)
- **NEAREST** (Rounding mode to round towards "nearest neighbor"). When both neighbors are equidistant, the RAISE mode is applied (e.g. 0.5 give 1.0, 0.4 give 0.0)

Charging Policy

The Chargeable Items Charging process can be launched specifying a rating mode which gives the possibility to tune the behavior of the process. This rating mode depends on multiple parameters such as:

- A number of CDRs
- A CDRs rating order
- A number of concerned subscription identifiers
- A CDRs retrieval mode
- And so on

8 different rating modes are implemented, and are grouped upon 3 implicit selection modes:

- **Day** mode, which is suitable for rating groups containing a small or a huge number of chargeable items, whatever the number of related subscriptions. **Day mode** corresponds to the **recommended setting** for all implementations. All other modes should be disregarded unless instructed otherwise by SAP for a specific use case
- **Subscription** mode, which is suitable for rating groups containing a small number of chargeable items related to a huge number of subscription identifiers
- **Windowed subscription** mode, which is suitable for rating groups containing a huge number of chargeable items related to a small number of subscription identifiers



NOTES

- The **subscription** and **windowed subscription** selection modes are slower than the **day** selection mode. Their use is deprecated and is thus not recommended to ensure global performances
- The charging policy only concerns the asynchronous and batch execution modes of the Chargeable Items Charging process

Subscription Mode

The **subscription** selection mode is used to retrieve all the chargeable items which are associated to a given batch rating group. These retrieved chargeable items are then grouped by subscription or provider contract identifiers, and transmitted to the Core Server in different threads:

- Sequentially, subscription or provider contract per subscription or provider contract
- In ascending order, using their consumption date

This selection mode is associated to the following rating modes:

Rating mode	Description
Selection by subscription and stop on first fail	In case of error during the charging of a chargeable item related to a subscription or provider contract, the BART Server stops the associated thread in order to avoid submitting the chargeable items whose consumption date is later than the consumption date of the erroneous chargeable item. The BART Server then sends the chargeable items relating to the other threads (i.e. the other subscription or provider contract identifiers)
Selection by subscription and rate all	All the chargeable items are processed even if an error occurs while a chargeable item relating to a subscription or provider contract is being charged. The BART Server then sends all the chargeable items relating to the same and the other threads
Selection by subscription and simulate	This mode is the same as the First Fail mode except the fact that the BART Server tries to charge the batch rating group without committing to the Core Server

Windowed Subscription Mode

The **windowed subscription** selection mode is used to retrieve all the chargeable items which are associated to a given subscription or provider contract identifier, in multiple times. The BART Server then performs the same operations as for the subscription mode.

This selection mode is associated to the following rating modes:

Rating mode	Description
Windowed selection and stop on first fail	In case of error during the charging of a chargeable item related to a subscription or provider contract, the BART Server stops the associated thread in order to avoid submitting the chargeable items whose consumption date is later than the consumption date of the erroneous chargeable item. The BART Server then sends the chargeable items relating to the other threads (i.e. the other subscription or provider contract identifiers)
Windowed selection and rate all	All the chargeable items are processed even if an error occurs while a chargeable item relating to a subscription or provider contract is being charged. The BART Server then sends all the chargeable items relating to the same and the other threads
Windowed selection and simulate	This mode is the same as the First Fail mode except that the BART Server tries to charge the batch rating group without committing to the Core Server

Day Mode

The **day** selection mode is used to process all the chargeable items, whatever the subscription or provider contract they are associated to. These retrieved chargeable items are grouped by consumption date, which requires fewer resources than the previous selection modes.

This selection mode is associated to the following rating modes:

Rating mode	Description
Selection by day and stop on first fail	In case of error during the charging of a chargeable item related to a subscription or provider contract, the BART Server filters the associated chargeable items (i.e. related to the same subscription or provider contract identifier) in order to avoid submitting the chargeable items whose consumption date is later than the consumption date of the erroneous chargeable item. The BART Server then sends the chargeable items relating to the other subscription or provider contract identifiers
Selection by day and rate all	All the chargeable items are processed even if an error occurs. The BART Server then sends all the chargeable items

Configuration Parameters

The Raters use several parameters located in a XML configuration file containing parameters related to SAP CC. For further information about these parameters, refer to the “Rater” section of the [SAP Convergent Charging Parameters Reference](#) documentation.

Process Functions

Rating Contexts Generation

The incoming external events contain the following consumption details:

- A service identifier, which represent the technical identifier of the consumed service
- A user service identifier (USID), which is a unique technical key representing an end customer within the marketable service
- A consumption date, used for comparisons with access chronology (an access referencing different activated charges of a subscription or a provider contract item at a different periods)

These properties are submitted to the internal “guiding” function, which gives the possibility to retrieve multiple information such as:

- The activation charge related to the event, and thus the associated subscription or provider contract
- The subscriber account

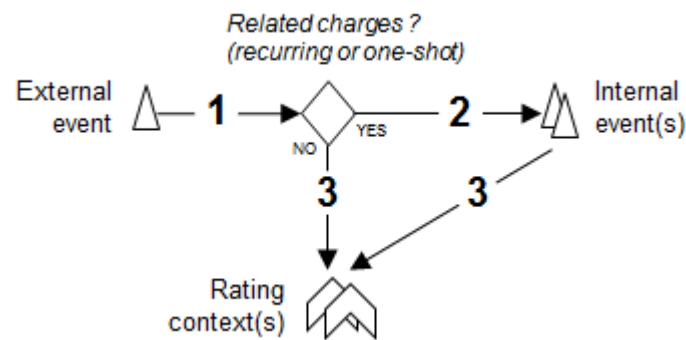
This information is then used by the Dispatchers to route the request to the adequate Rater in charge of charging the external event.

The first step performed by a Rater consists in executing the Activation process (step 1) to ensure that all recurring and/or one-shot internal events related to the external event have been triggered. Then, according to the result of this Activation process execution, some internal events may be generated (step 2). Finally, all the handled events are materialized by rating contexts containing all the information necessary for rating purposes (step 3).



NOTE

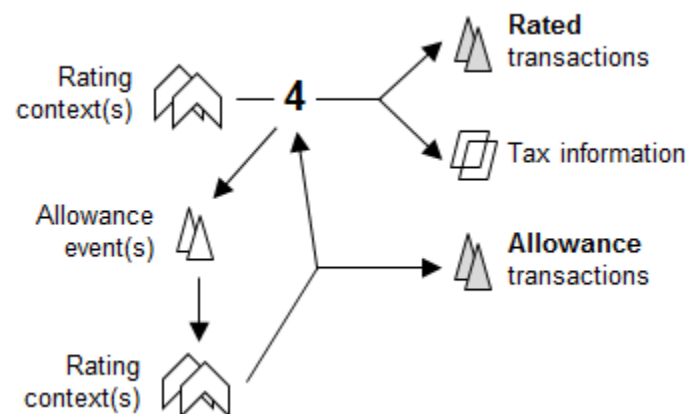
One or more rating contexts can be built according to the defined charges. The initial master charge is used as an entry point to build the rating context(s) associated to the dependent charge(s).



Events Pricing

Once all the rating contexts have been created and updated, the Rater executes the different price plans which are associated to the events (step 4) in order to:

- Calculate all the adequate rated amounts (prices and so on), using the concerned allowances when necessary
- Create the corresponding transactions, which are then considered as “rated transactions”, and their associated tax information



NOTE

Allowance events are sent during the calculation algorithm of the charge, when executing the price plan interacting with the allowances. The allowance events which are updated by the allowances are available for further processing operations.

Events Charging

Once all the rated transactions have been created, the Rater retrieves the tax settings set in the related subscriber account and updates the tax information (step 5) determined during the execution of the price plans accordingly. This updated tax information is then used to calculate the adequate tax data (step 6).

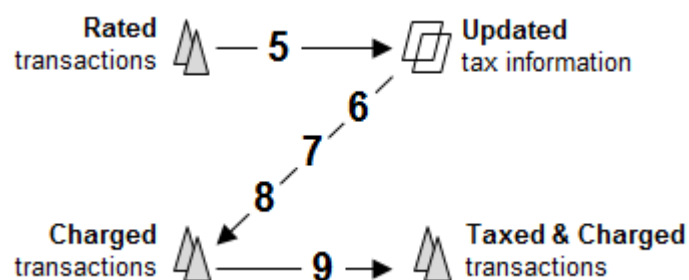
Once the taxes have been computed, the Rater executes the charging plan related to the concerned charges (step 7) in order to identify all the concerned subscriber account balances (which may include linked account balances in case of overrun), which can refer to both prepaid balances and external balances. Every rated transaction is split in charged transactions relating to each identified balance (step 8). Each charged transaction contains a base amount

so that the sum of the base amounts of all the charged transactions referring to the same rated transaction, is equal to the rated amount.

The computed taxes are then adjusted and applied (or not) on these charged transactions (step 9), which thus become taxed & charged transactions, ready for persistency purposes.

These balances are then updated (step 10) by the Rater in order to take into consideration the consumption of services. This modification:

- Is immediately performed if the transaction refers to a prepaid balance or credit-limit balance
- Depends on the deployed third-party billing system if the transaction refers to an external balance:
 - No modification is performed on the account balance if no third-party billing system is deployed
 - A dedicated function is executed if a third-party billing system is deployed

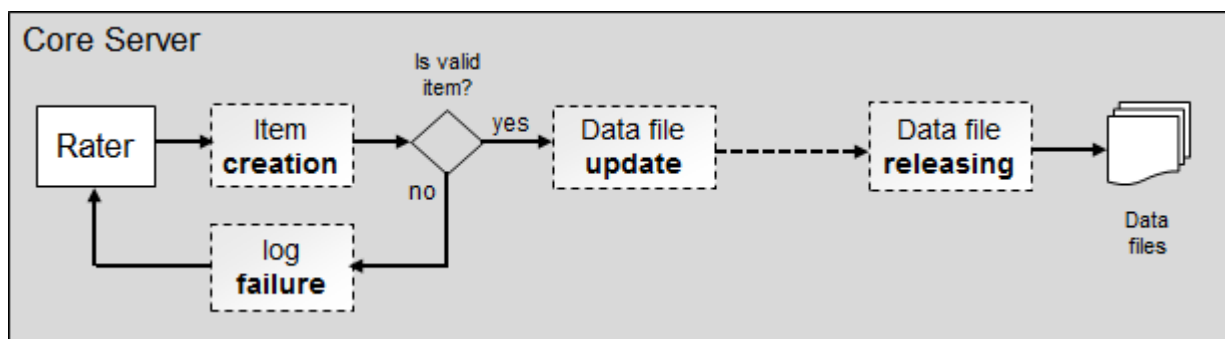


Allowance Creation

As allowances are associated to provider contracts, the Allowance Creation function is executed after the Events Charging function (which is used to retrieve the information related to the account, and thus to the provider contract). This function consists in creating the allowance that has been provisioned during the calculation algorithm, which implicitly executes the one-shot trigger components declared in the concerned allowance logic objects (and which relate to a creation event).

Data Files Generation

The Data Files Generation function of the Chargeable Items Charging process consists in validating the taxed & charged transactions and recording them into dedicated data files to give the possibility to a third-party system to exploit this information.



By default, the Raters try to validate each item by using the associated billable item mapping. When an item is considered as invalid, the corresponding charging request is rejected and the targeted data file is not updated. An

error message is written in a log file, to inform about the reason of the validation failure, which can be due to the following main reasons:

- The value of an item field is incorrect regarding the mapped billable item field, because:
 - The String type value has too many characters
 - The String type value is not a value of the String type enumeration like Currencies, Sub Processes or BIT Types
 - The Number type value has too many digits
 - The Number type value has too many decimal places
 - The Number type value is negative but only a positive value is expected
- A mandatory billable item field is mapped to an item field that is not set
- The value of the BIT Type field is not assigned to the given Sub Process
- It is not possible to convert a decimal number into an Amount type field because there is no currency defined in the system with enough decimal places

Every item considered as valid regarding the used mapping is recorded in the targeted data file for further bulk loading operations. This persistency mechanism is managed through a dedicated framework which gives the possibility to implement:

- A **Transaction File** processor, which creates a single file whose content can be customized
- A **Data File** processor, which creates multiple data files which contain different data related to transactions
- A **Charged Item File**, a **Refill Record File** or a **Chargeable Item File** processors, which give the possibility to create data files whose content corresponds to taxed & charged transactions transcribed into charged items, refill records or chargeable items (using mappings defined at offer, charge plan, refill plan, or allowance plan levels)
- A **Rerate File** processor, which gives the possibility to create data files whose content can be used by a third-party system for rerating purposes



NOTE

The "TRANSAC_PROC_CLASS" parameter gives the possibility to switch between these processors. For further information about this parameter, refer to the [SAP Convergent Charging Parameters Reference](#) and [SAP Convergent Charging Configuration Guide](#) documentations.

Transaction File and Data File processors

The use of Transaction File and Data File processors is deprecated. To ensure compatibility with offers created through previous versions of SAP CC, it is still possible to use them. For further information about these processors, refer to older documentation available in SAP Help Portal.

Charged Item File, Refill Record File or Chargeable Item File processors

All these processors are dedicated to output data coming from a charging context and intended to be loaded into a third party billing system or another system. They implement a common output framework named Charging output Integration Framework (CIF) to output:

- Chargeable items, during the execution of the [Chargeable Items Acquisition](#) and [Chargeable Items Charging](#) processes
- Charged items, during the execution of the [Chargeable Items Charging](#) and [Prepaid Accounts Refilling](#) processes
- Refill records, during the execution of the [Prepaid Accounts Refilling](#) process

These processors enable to:

- Filter charged items, chargeable items, or refill records depending on the value of an item field or depending on the class of an item
- Record these data into files with the same CSV format.

You can configure these processors to meet specific needs and tune the writing policy parameters such as:

- The writing location
- The number of writing channels
- The file rolling policy
- The file compression (using the RFC 1951 specifications)
- The activation of the automatic cleaning mechanism which is used to remove:
 - The uncommitted items from data files
 - The associated control file



NOTES

- When the file compression is enabled, the data file may not contain all the items in case of instance failure
- When the automatic cleaning mechanism is enabled, if an error occurs during the cleaning operation then the data file is set in error by adding the “.cleanErr” extension to its filename

The Raters can generate data files using 2 different formats:

- Version 1, which only contains the items
- Version 2, which is available from SAP CC 4.0 SP07 and contains additional information

Version 1 data files

The filenames of the version 1 data files are made up with the following information, each element separated by a underscore (_) sign:

- CIT class name
- Writing channel identifier (3 digits number)
- File rolling policy (3 characters string such as “HOU” for hourly rolling policy)
- Writing channel index (2 digits number)
- Timestamp using the yyyyMMddTHHmssSSS format (e.g. 20141101T123456789)

The version 1 data files contain:

- One and only one header block containing the names and types of the fields which are contained in the data file
- One or several items, written using a CSV format

Version 2 data files

The filenames of the version 2 data files are made up with the following information, each element separated by a underscore (_) sign:

- If the “fileSequenceId” attribute is empty or missing:
 - CIT class name
 - Writing channel identifier (3 digits number)
 - File rolling policy (3 characters string such as “HOU” for hourly rolling policy)
 - Writing channel index (2 digits number)
 - Timestamp using the yyyyMMddTHHmssSSS format (20141101T123456789 for example)
- If the “fileSequenceId” attribute is not empty:

- SAP system identifier (3 alphanumeric characters string)
- CIT class name
- Writing channel identifier (3 digits number)
- File rolling policy (3 characters string such as "HOU" for hourly rolling policy)
- Writing channel index (2 digits number)
- Timestamp using the yyyyMMddTHH:mm:ssSSS format (20141101T123456789 for example)
- The sequence number is a 6-digit one (range 000000-999999)

The version 2 data files contain:

- One and only one header block containing:
 - The version of the data file
 - The name of the corresponding class
 - The names and types of the fields contained in the file
- One or several items, written using a CSV format



NOTES

- Version 2 is enabled by default for new fresh installations. During an upgrade of a SAP Convergent Charging system, this version is not updated, to ensure backward compatibility and preserve the potential already running integration with external systems dealing with data files
- To modify the version of the data files, refer to the [SAP Convergent Charging Configuration Guide](#) documentation

Whatever the version is, the content of the data files respect the following rules:

- All files use the same data separator: “,” (comma)
- If the line numbering flag is set to true, the first field of each line corresponds to the line number
- Every line contains the properties of a charged item (or refill record)
- Every line is terminated by the “\n” line break command
- Details containing commas are surrounded with double-quotes
- Details containing line-breaks are surrounded with double-quotes
- Double-quotes located inside details are escaped by a double-quote
- Dates and times are formatted using ISO 8601 format
- Numbers are formatted using the default Java `BigDecimal.toString()` method
- Strings are not formatted



NOTES

- For further information about the configuration of the processors, refer to the [SAP Convergent Charging Configuration Guide](#) documentation
- Either Charged Item File or Refill Record File processors **must be used** when offers are created within the SAP CC 2.0 release. In case of SAP CC 1.0 and 2.0 mixed offers, this processor automatically implements the Transaction File processor to handle the SAP CC 1.0 offers
- These processors behaves as data interceptors, and are thus able to filter, replace or validate data before sending them into data files. These features are not provided in the default provided processors, and thus require specific developments
- These processors generate a control file for each generated data file. A control file contains the encoded identifier of the last element written in the associated data file, which gives the possibility to follow and track

the writing operations performed by the Rater server instances. For further information about control files, consult the section related to the [Data Files Bulk Loading process](#)

Rerate File processor

The Rerate File processor represents an implementation of the TIF used for rerating purposes. It gives the possibility to record data files which:

- All have a CSV format
- All have the same content which is used to build rerating commands on given subscriptions

Like the Charged Item File processor, the Rerate File processor can be configured to fit specific needs and tune the writing policy parameters such as:

- The writing location
- The number of writing channels
- The file rolling policy
- The file compression (using the RFC 1951 specifications)

The generated files respect the following rules:

- The filename is made up with the following information separated by a underscore sign (_):
 - The "RERATE" string
 - Writing channel identifier (3 digits number)
 - File rolling policy (3 digits string such as "HOU" for hourly rolling policy)
 - Writing channel index (2 digits number)
 - Timestamp using the yyyyMMddTHHmssSSS format (20100501T145723256 for example)
- All files use the same data separator: "," (comma)
- The first line acts as a header and contains the list of properties, using a **name:type** format (SUAC_CODE:string, FROM_DATE:date, and so on). This line is terminated by the "\n" line break command. The following properties are written in each generated data file:
 - **SUAC_CODE**, a string which corresponds to the subscriber account code
 - **SUB_CODE**, a string which corresponds to the subscription code
 - **FROM_DATE**, a date which corresponds to the rerating date provided in the rerating screen of the Core Tool user interface
 - **FROM_CITS_ID**, a number which represents the lower bound of the list of charged item sets concerned by a deletion operation
 - **TO_CITS_ID**, a number which represents the upper bound of the list of charged item sets concerned by a deletion operation
- Every line contains the values of each property used to build a rerating command, and ends with the "\n" (line break) command
- Details containing commas are surrounded with double-quotes
- Details containing line-breaks are surrounded with double-quotes
- Double-quotes located inside details are escaped by a double-quote
- Dates and times are formatted using ISO 8601 format
- Numbers are formatted using the default Java BigDecimal.toString() method
- Strings are not formatted

When handled by a third-party billing system, these data files give the possibility to identify exhaustive lists of charged items which must be impacted in case of rerating operations (deletion, deactivation, and so on according to the needs).

**NOTES**

- For further information about the configuration of the Rerate File processor, refer to the [SAP Convergent Charging Configuration Guide](#) documentation
- When rerating operations must be performed, it is possible to use both Data File and Rerate File processors. As the use of the Data File processor is deprecated, **it is highly recommended to use the Rerate File processor**

Immediate Threshold Refilling

The Immediate Threshold Refilling function is the last executed function of the Chargeable Items Charging process. It consists in re-analyzing the processed charges in order to determine whether they are associated to a refill plan or not. If a refill plan exists and contains some account event refills, the Rater determines whether it has some immediate refills to trigger or not. To do so, it compares the amount of the prepaid account balances to the defined thresholds, and immediately triggers all the refilling operations which match the condition.

**NOTES**

- Immediate threshold refills correspond to refilling operations:
 - Related to prepaid account balances only
 - Defined into a refill plan using the Account Event Refill rate component. As refill plans refer to provider contracts, these refilling operations are thus not available when using subscriptions
- To avoid any latency during a real-time charging operation, immediate threshold refills are performed after the Data Files Generation function, in a different database transaction

*Execution Modes***Online**

The Chargeable Items Charging process can be executed by the Raters in an online mode. This execution mode is used to process in a real-time each incoming external event one after the other, and can be considered as stateful or stateless, according to data persistency considerations.

The default execution mode corresponds to the stateful one, when SAP CC is connected with third-party applications such as a mediation system, a credit-control system, and so on.

When the stateless mode is used, the rating engine can be considered as a simple computing unit, which implies particularities and restrictions such as:

- The processed external events must only relate to a master charge (retrieved with a provided charge code)
- A predefined rating context must be provided to the charging operation to specify information about counters, parameters, and translation/tier tables (but without any information related to offer or charge plan)
- No customer master data is stored, which implies that the client application is responsible for the storage of counters and input data
- No data files are generated at the end of the charging operation
- Taxes cannot be managed
- Detailed information related to transactions are not filtered
- Only charges referring to subscriptions can be processed

**NOTE**

Offers or charge plans information provided in the specified rating context may contain dependencies between the associated charges. These possible dependent charges cannot be managed when using the stateless mode, as only a single master charge is provided to the charging operation.

External & Internal Events

When using the stateful mode, the received **external** events (chargeable items) lead to the production of **internal** recurring or one-shot events. When using the stateless mode, these internal events become external ones, whose activation leads to a modification of the initially provided stateless rating context.



NOTE

Managed types of events are:

- Usage
- Activation
- Suspension
- Resumption
- Cancelation *
- Early Cancelation *

** not available for convergent charging services based on provider contracts*

Stateless Rating Context

Stateless rating operations require the following information:

- The identification code of a master charge configured in SAP CC, which is mandatory as used to retrieve the related pricing logic
- A stateless rating context, which is mandatory and contains information related to the rating operation and to the customer master data
- A creation date, which is mandatory and corresponds to the subscription date
- The last rating date, which is optional and automatically set to the creation date when not provided. When provided, this date must be greater than the creation date

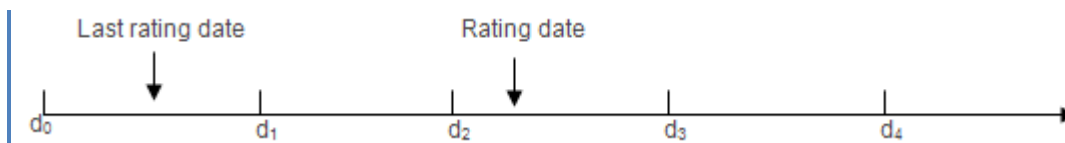
The first step of a stateless rating operation consists in analyzing the provided consumption information in order to trigger the Activation process. The provided creation date is then analyzed to determine:

- The adequate version of the price plan to use for pricing purposes (which corresponds to an element of the price plan chronology)
- A period which potentially contains a list of usage, recurring and one-shot events which may be triggered, and which may differ according to the triggering type of event:
 - At the beginning of a period
 - At the end of a period



EXAMPLE

Considering the following timeline:



If an event is associated to the d_1 date, the computed period can be:

- (d_1-d_0) if the event must be triggered at the end of a period
- (d_1-d_2) if the event must be triggered at the beginning of a period

According to the type of rated event, the following events can be generated:

Event type	Triggered elements
Usage	<ul style="list-style-type: none"> ■ One-shot rates (CREATION subtype) are triggered if the last rating date is null ■ Recurring rates are triggered between the last rating date (or the creation date if the last rating date is null) and the rating date ■ Usage rate is computed
Activation	<ul style="list-style-type: none"> ■ One-shot rates (CREATION subtype) are triggered if the last rating date is null ■ Recurring rates are triggered between the last rating date (or the creation date if the last rating date is null) and the rating date
Suspension	<ul style="list-style-type: none"> ■ One-shot rates (CREATION subtype) are triggered if the last rating date is null ■ Recurring rates are triggered between the last rating date (or the creation date if the last rating date is null) and the rating date ■ One-shot rates (SUSPENSION subtype) are triggered at the rating date
Resumption	<ul style="list-style-type: none"> ■ Recurring rates charged at the beginning of the period are charged once at the rating date (prorated if necessary) ■ One-shot rates (RESUMPTION subtype) are triggered at the rating date
Cancellation	<ul style="list-style-type: none"> ■ One-shot rates (CREATION subtype) are triggered if the last rating date is null ■ Recurring rates are triggered between the last rating date (or the creation date if the last rating date is null) and the rating date ■ One-shot rates (CANCELLATION subtype) are triggered at the rating date
Early Cancellation	<ul style="list-style-type: none"> ■ One-shot rates (CREATION subtype) are triggered if the last rating date is null ■ Recurring rates are triggered between the last rating date (or the creation date if the last rating date is null) and the rating date ■ One-shot rates (EARLY CANCELLATION subtype) are triggered at the rating date

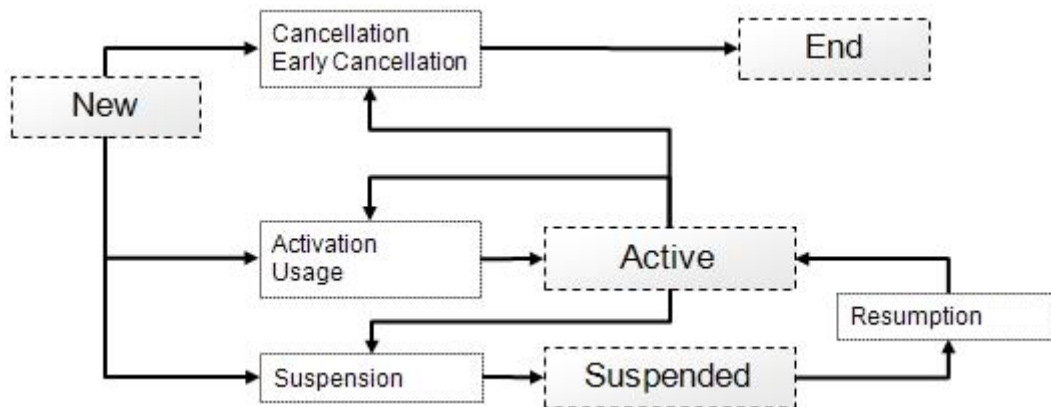
The second step of a stateless rating operation consists in applying every adequate price plan on each event. The content of the stateless rating context is then analyzed to get the values of counters, parameters, and translation tables.

NOTE

If the rating context contains objects whose values are not initialized, the default values defined in the charge component are used.

According to the type of concerned event, the state of the stateless rating context is implicitly modified to a new state which corresponds to a virtual state, as described below:

Event type	Dates meaning	State transition	
Usage	Rating date = date of the usage	Before	Active or New
		After	Active
Activation	Rating date = date of the activation	Before	Active or New
		After	Active
Suspension	Rating date = date of the suspension	Before	Active or New
		After	Suspended
Resumption	Rating date = date of the resumption Last rating date = suspension date	Before	Suspended
		After	Active
Cancellation	Rating date = date of the cancellation	Before	Active or New
		After	End
Early Cancellation	Rating date = date of the cancellation	Before	Active or New
		After	End



Offline

When your business does not require real-time execution and credit-control operations, it is possible to execute the Chargeable Items Charging process in an offline (stateful) mode.

This execution mode can either deal with single external events or with a list of external events, but due to its disconnected characteristic, this execution mode is generally associated to batch charging operations which can be performed:

- Directly by the Core Server system, taking into account a list of external events provided by a client application. This corresponds to the online stateful mode, except the fact that:
 - A sorted list of external events is provided for charging purposes, instead of a single event
 - The provided list of external events only contains events relating to the same subscription or provider contract (the operation fails if an event relates to another subscription or provider contract than the other events). To avoid such a failure, a "subscriptionID" argument can be specified to limit the processing scope of the operation
 - The operation can be launched using a rating mode which modifies the behavior of the process execution, in terms of chargeable items retrieval and order, failure management, and so on. For further information about this rating mode parameter, refer below
- By the BART Server system (for better performances), which deals with batch charging sessions:
 - Stored in its associated BART database for tracking (and possibly rerating) purposes

- Associated to a given batch rating group identifier, used to group a set of chargeable items and related to one or more subscriptions or provider contracts
- Launched using a rating mode which modifies the behavior of the process execution, in terms of chargeable items retrieval and order, failure management, and so on. For further information about this rating mode parameter, refer below

**NOTES**

- The chargeable items which are stored in the BART database both concern prepaid and external accounts, but only chargeable items referring to external accounts are taken into account in case of subsequent rerating operations
- Grouping chargeable items on a batch rating group identifier implies that these chargeable items can be taken into account in an order which can be different from their consumption date. As a consequence, only subscriptions referring to external balances can be rated in this batch mode, in order to ensure data integrity in case of subsequent rerating operations (as prepaid balances could be incorrectly computed and thus rejected, which would lead to data inconsistency)

The disconnected characteristic of the offline stateful execution mode gives the possibility to safely interrupt a charging operation, for example in case of peak period.

Inverse

The Chargeable Items Charging process can be executed in an “inverse” mode. This mode is used to convert a sum of money (i.e. the balance of a prepaid account) into a possible usage quota according to a given price plan and an existing stateful context (containing counter values, parameters, and so on).

This process is based on a dichotomy algorithm which is mixed with an “ax+b” approach in order to converge towards the solution rapidly. To ensure performances, a maximum of 500 iterations is allowed for the dichotomy algorithm. If the concerned price plan is not able to converge in less than 500 iterations, an error is returned.

**EXAMPLE**

“How long can Mr. BAR call Tahiti with \$5, in the limit of 60 minutes, after having subscribed the FOO call offer?”

The answer to this question depends on the subscribed offer and depends on the Mr. BAR potential counter values:

- If the offer includes, for example, a 1 hour free call balance per month, the answer will also depend on this current balance value
- If the FOO offer is based on a model with 50 cents/min and 1 hour free calls per month and if the balance current value is set to 15 minutes, the customer will be able to call Tahiti during 10 + 15 (free) minutes

The chargeable item which is sent to an “inversed” charging operation must contain a value for the property to inverse. This value represents the maximum quota that can be returned for the property to inverse, and must be valid regarding the defined precision. This value is supposed to be between 0 and the specified maximum quota.

This execution mode can manage the precision related to the service unit (quota). Most of the time, this precision is related to a discrete quantity and may be set, for example to 1 (but a different value is allowed).

**NOTE**

When the price plan depends on the property to inverse, the “inversed” charging operation can return a value which does not correspond to the exact limit, and can thus fail when really charged.

Best-Effort

The Chargeable Items Charging process can be executed in a “best-effort” mode, which is used to charge an account with the maximum quantity of available service. This execution mode is made up with 2 successive steps:

- An initial “inversed” charging operation, which is used to determine the maximum quota of service available for charging purpose
- An immediate charging operation, which directly debits the determined quota of service (even if the service is not delivered)



NOTE

The session-based execution mode also contains a “best-effort” mechanism, which concerns the reservation of service quotas. Do not confuse best-effort charging operations and best-effort reservations made during session-based charging operations. For further information about the best-effort reservation mechanism, refer afterwards.

Blank

The Chargeable Items Charging process can be executed in a “blank” mode, which guarantees that the databases stay unmodified after the execution of the related charging operation. Using this mode, users can communicate with SAP CC to perform charging operations for advice of charge purposes.

Session-Based

Process Description

A charging session (such as phone call or video streaming) is considered as a succession of simple events, split into quotas of customer service which must be charged, and occurring during a limited period. SAP CC gives the possibility to manage operations relating to:

- Mono service sessions, where each session concerns a single customer service
- Multiservice sessions, where a given session can refer to multiple customer services (each consumed service referring to a given mono service session)

The session-based execution mode of the Chargeable Items Charging process is similar to the Online (stateful) execution mode, with additional features such as:

- History mechanisms used to record usage quotas and potentially associated counters reserved at session start and/or session update times
- Session update operations, which represent intermediate interim rates used to partially or completely confirm previous reserved elements and make new reservations of usage quotas when necessary
- Session stop rate, which represents a final rate used to partially or completely validate the previously reserved service usage quotas
- Best-effort reservation mechanism, which gives the possibility to compute and reserve a maximum quota of service in case of standard reservation failure. This mechanism always concerns a unique service unit (as only one property can be inversed), and is activated when specifying the “propertyToInverse” argument at session start time
- Charged Items Aggregation, which gives the possibility to apply specific rules on the different properties of the charged items in order to merge these charged items and only keep a pending one in a session

- (Optional) reservation renewal mechanism, which gives the possibility for a client application to reserve again a quota of service in a charging session which is inactive due to insufficient credit



NOTES

- SAP CC supports partial confirmation of credit reservations. The configuration of the price plans executed during the session-based charging operations must ensure that the prices computed for a complete confirmation are greater or equal to the prices computed for a partial confirmation
- As of SAP CC 5.0 SP 3, it is possible to control the maximum number of charging sessions that can be started for every subscriber account. For further information, refer to the “Limiting the number of charging sessions” section of the [SAP Convergent Charging Tuning Guide](#) documentation

The session-based execution mode supports credit-control and quota reservation in real-time. When credit-control is not implemented, charging a session is equivalent to charging the event generated at the end of this session. Quota reservations are grouped for each provider contract or subscription into a charging session history, based on a charging session per session. A quota reservation includes at least the last consolidated chargeable item and its related reserved counter(s). For a given multiservice session, the session context contains:

- The accumulated amounts corresponding to the sums of the amounts to confirm computed by the mono service sessions. There is one amount per currency
- A reservation renewal listener (RRL) identifier (optional)
- Some information dedicated to the notification listener in the registered client application or system

Transient Counters

As described in the Master Data section of this documentation, transient counters represent temporary counters associated to a charge component or to an allowance logic. These counters are not visible and cannot be redefined in offers, charge plans or allowance plans containing the related charge or allowance logic (but they can be shared between a master charge and its dependent charges). Transient counters also give the possibility to use stepped price plans where consumption cost may vary according to the total consumption.

The session-based execution mode uses transient counters to aggregate all the confirmed service usage quotas. They are both saved into the database and stored in memory as cached data (in the session rating cache). The life cycle of transient counters is linked to a charging session, as they are created when the corresponding charging operation starts. More precisely, the transient counters of the master charge are created first and then propagated to the dependent charges. If some transient counters expected by the dependent charge do not match the provided ones (name correspondence), they are created. When the session-based charging operation is stopped, its associated transient counters are deleted.

Transient and permanent counters are both declared and described using the same price plan component which gives the possibility to specify:

- The name of the counter, which must be unique within a given price plan
- A description, which describes the role of the counter
- The default value of the counter, which is used at creation time



NOTES

- Transient counters have an additional option which gives the possibility to enable or disable their update during the reservation operations. When this option is disabled for a given transient counter, its value is not updated during the reservation operation, which thus corresponds to a behavior similar to persistent counters

- When configured, a transient counter can be used within all the rate components of a price plan whatever their type in the reusable charge (one-shot, recurring or usage)
- When the declaration of transient counters is the same for a master charge and its dependent charge, the value of this counter is shared at runtime

Transient counters are owned by only one session and cannot be shared with other sessions. Transient counters life cycle is thus linked to the session which owns them:

- At session start time, transient counters are created (with their specified default value) and stored in both memory (in the session cache) and database
- At session update time, values of transient counters are modified and updated. If a transient counter is impacted by a reservation operation, this impact is only visible at the next confirmation (contrary to persistent counters which are immediately updated)
- At session stop time (either during confirmation or cancellation operations), transient counters are deleted from the session

Session History


The sessions related to a provider contract item or to a subscription are grouped into a **session history**. The life cycle of this history depends on the life cycle of its related contract item or subscription:

- When a related contract item or a subscription is deleted, the related session history is also deleted. Pending sessions are thus lost and future update and stop operations related to these sessions will fail.
- A contract item or a subscription can be modified by removing a charge activation. All the sessions managed through a deleted charge activation are removed and related counter reservations are cancelled.
- All the sessions managed through a deleted access are removed and related counter reservations are cancelled

Mono service Sessions

The following table lists the different operations which can be performed on a mono service session:

Operation	Description
Start	<p>When a mono service session starts, the rating engine is called with the following information:</p> <ul style="list-style-type: none"> ■ A unique external session identifier ■ A service identifier and a user service identifier (USID), which together define an access ■ A start date, which is combined with the above access to retrieve the adequate subscription or provider contract item ■ A chargeable item, which includes the requested quota of service per property and is used for reservation purposes ■ A rating result type, which is used to filter the transactions returned at the end of the operation. Possible values for the rating result type are: <ul style="list-style-type: none"> ● Usage transactions ● Usage, recurring and one-shot transactions ● Master usage transactions ● No transaction ■ A time to live, which defines the session expiration time ■ A default resolution, which determines the behavior of the session at expiration time <p>Then, the rating engine:</p> <ul style="list-style-type: none"> ■ Charges the provided chargeable item

	<ul style="list-style-type: none"> ■ Performs the adequate quota reservations if required and/or necessary ■ Creates a session, keeping the original chargeable item used for reservation purpose ■ Creates a chargeable item by comparing: <ul style="list-style-type: none"> ● The reservation chargeable item stored in the session ● The provided chargeable item potentially empty, partially or completely filled <p> NOTES</p> <ul style="list-style-type: none"> ■ This reservation chargeable item is rated and concerned counters are reserved: <ul style="list-style-type: none"> ● Systematically for persistent counters ● Potentially for transient counters according to their configuration (see above for explanation about transient counters update during reservation) ■ The previously created rating result is enhanced and contains: <ul style="list-style-type: none"> ● Optional transactions used to reserve quota for the current reservation ● An amount to reserve. The amount to reserve is the sum of the amounts contained in the above-mentioned transactions to reserve.
Update	<p>When a mono service session is updated, the rating engine is called with the following information:</p> <ul style="list-style-type: none"> ■ The unique identifier of the session, which corresponds to the one provided at start time ■ A service identifier and a user service identifier (USID), which together define an access ■ An update date, which must correspond to the start date in order to match the correct access chronology ■ A rating result type, which is used to filter the transactions returned at the end of the operation. Possible values for the rating result type are: <ul style="list-style-type: none"> ● Usage transactions ● Usage, recurring and one-shot transactions ● Master usage transactions ● No transaction returned ■ An optional chargeable item, which includes the used quota of service per property and which is used for confirmation purposes. When this chargeable item is not provided, the system automatically confirms the previously reserved quota of service. ■ An optional chargeable item, which is used for reservation purposes. When this chargeable item is not provided, the system automatically reserves the same quota of service than the one specified in the previous reservation. This chargeable item includes: <ul style="list-style-type: none"> ● The requested quota of service per property ● A list of updated properties (for example the QoS during a video conference) <p>First of all, the rating engine retrieves the concerned session and charges again the reservation chargeable item stored in this session. A rating result is then created and contains:</p> <ul style="list-style-type: none"> ■ Optional transactions (*) used to cancel the previous reservation operation which impacted the related account balance ■ An amount to cancel. The amount to cancel is the sum of the amounts contained in the above-mentioned transactions to cancel <p>Secondly, the rating engine creates a consolidated chargeable item by comparing:</p> <ul style="list-style-type: none"> ■ The reservation chargeable item stored in the session ■ The confirmation chargeable item potentially provided (or automatically re-created)


NOTES

- The consolidated chargeable item is rated and previously reserved counters are updated in both database and memory
- The previously created rating result is enhanced and contains:
 - Optional transactions used to **confirm** the previous reservation operation which impacted the related account balance
 - An amount to confirm. The amount to confirm is the sum of the amounts contained in the above-mentioned transactions to confirm
- The transactions stemmed from this rating result are sent to the TIF and written into the adequate data files

Thirdly, the rating engine creates a consolidated chargeable item by comparing:

- The reservation chargeable item stored in the session
- The provided chargeable item potentially empty, partially or completely filled


NOTES



- This new reservation chargeable item is rated and concerned counters are reserved:
 - Systematically for persistent counters
 - Potentially for transient counters according to their configuration (see above for explanation about transient counters update during reservation)
- The previously created rating result is enhanced and contains:
 - An amount to reserve
 - Optional transactions used to reserve quota for the current reservation

Finally, the rating engine updates the session with this new consolidated reservation chargeable item considered as the new reference for the next session update operation.

Stop

When a mono service session is stopped, the rating engine is called with the following information:

- The unique identifier of the session, which corresponds to the one provided on session start and thus on session update times
- A service identifier and a user service identifier (USID), which together define an access
- A rating result type, which is used to filter the transactions returned at the end of the operation. Possible values for the rating result type are:
 - Usage transactions
 - Usage, recurring and one-shot transactions
 - Master usage transactions
 - No transaction returned
- A default resolution, which determines the session resolution at expiration time (confirmation, cancellation, and so on)
- An optional chargeable item, which includes the requested quota of service per property and which is used for **confirmation** purposes. When this chargeable item is not provided, the system automatically confirms the previously reserved quota of service.

	<p>First of all, the rating engine retrieves the concerned session and rates again the reservation chargeable item stored in this session. A rating result is created and contains:</p> <ul style="list-style-type: none"> ■ An amount to cancel ■ Optional transactions (*) used to cancel the previous reservation operation which previously impacted the related account balance <p>Secondly, the rating engine creates a consolidated chargeable item by comparing:</p> <ul style="list-style-type: none"> ■ The reservation chargeable item stored in this session ■ The confirmation chargeable item potentially provided <p> NOTES</p> <ul style="list-style-type: none"> ■ The consolidated chargeable item is rated and previously reserved counters are updated into both database and memory ■ The previously created rating result is enhanced and contains: <ul style="list-style-type: none"> ● An amount to confirm ● Optional transactions used to confirm the previous reservation operation which previously impacted the related account balance ■ The transactions stemmed from this rating result are written into adequate data files <p>Finally, the session is deleted and the session identifier becomes available for next sessions. If the “resolution” argument is set to CANCELLED:</p> <ul style="list-style-type: none"> ■ The counter reservations coming from the last reservation (from session start or session update) are cancelled ■ The created rating result may be empty
Cleanup	<p>In case of service failure such as equipment failure, a session stop operation can be lost. When such a situation occurs, unclosed pending sessions can exist and thus require a specific action in order to clean those pending sessions. An automatic cleaning policy can be configured session per session according to the “ttl” (time to live) argument provided at session start.</p> <p> NOTE</p> <p>To clean a session, it is also possible to confirm or cancel the last counter reservations using the “defaultResolution” argument provided at session start.</p> <p>This session cleaning function is both triggered:</p> <ul style="list-style-type: none"> ■ At the beginning of the start, update and stop operations, on targeted subscriber account ■ Periodically during the cleaning operation of the deprecated Prerating and Postrating process, on all subscriber accounts (when this process is scheduled)

(*) The generation of the transactions to cancel is deactivated by default for performance reasons, but can be activated using the TRANSACTIONS_TO_CANCEL_IN_RESULT_ENABLED parameter. The generation of the transactions to cancel must be activated if balances are handled by an external system. If the generation of the transactions to cancel is deactivated, the amount to cancel is not computed. For further information about the TRANSACTIONS_TO_CANCEL_IN_RESULT_ENABLED parameter, refer to the [SAP Convergent Charging Parameters Reference](#) documentation.

 **NOTE**

The reservation renewal mechanism is not available with mono service sessions. For further information about this mechanism, refer afterwards.

Chargeable Item Consolidation

Considering a chargeable item including user properties which can be used by rating components, for example:

```
Name: call
Property: event (the session event described as a string)
Property: QoS (the quality of service described as a string)
Property: duration (the call duration described as a number)
```

It is necessary to consolidate the chargeable item during the update and stop operations, in order to create a complete consolidated chargeable item. This operation is performed thanks to the comparison between:

- The properties defined in the reservation chargeable item
- The properties defined in the confirmation chargeable item

If some properties are not present in the confirmation chargeable item or when it is not provided due to its optional characteristic, they are added to the consolidated one.

 **NOTE**

This consolidation operation is also required to create a complete new reservation chargeable item at session update time. It is then saved into the session and available for next update operations.

 **EXAMPLE**

The following example illustrates the consolidation operation performed on a chargeable item described above. In this example, the QoS property is only provided at session start time:

			Provided	Saved	Consolidated
Start		→	Reservation: call(event = 'start', QoS = 'good', duration = 3)		call(event = 'start', QoS = 'good', duration = 3)
	OK	←		call(event = 'start', QoS = "good", duration = 3)	
Update	Stop	→	Confirmation: call(duration = 2)		call(event = 'start', QoS = 'good', duration = 2)
	OK	←			
	Start	→	Reservation: call(event = 'update', duration=4)		call(event = 'update', QoS = 'good', duration = 4)
	OK	←		call(event = 'update', QoS = 'good', duration = 4)	
Update	Stop	→	Confirmation: call(duration = 1)		call(event = 'update, QoS = 'good', duration = 1)
	OK	←			
	Start	→	Reservation: call()		call(event = 'update', QoS = 'good', duration = 4)

	OK	←		call(event = 'update', QoS = 'good', duration = 4)	
Stop (3 min used)		→	Confirmation: call(event = 'stop', duration = 3)		call(event = 'stop', QoS = 'good', duration = 3)
End		←		<i>Deleted from cache</i>	

At session update time, a first consolidation is made before rating the confirmation chargeable item. A second consolidation is also made to save a complete reservation chargeable item into the session.

Immediate Threshold Refilling

The immediate threshold refilling operation is performed at the end of the update, stop and cleaning operations of every mono service session. For further information about immediate threshold refilling, refer above.

Charged Items Aggregation

When a mono service session deals with charged items concerned by an aggregation policy, the behavior of the start, update and stop operations is modified as follows in order to handle the aggregation of these charged items:

- If the aggregation policy deals with an aggregation period, the time of the **start** operation is used as the initial start time of this aggregation period
- Every **update** operation refers to a chargeable item which is used for confirmation purpose. This chargeable item is processed and the generated charged item is aggregated with a pending charged item (which can correspond either to the previously aggregated one or to a null element when the update operation corresponds to the first update one)
- The behavior of the **stop** operation depends on the resolution of the session:
 - A stop operation used to cancel the session does not perform any confirmation operation, and thus only leads to the reporting of the pending charged item(s) into the external billing system
 - A stop operation referring to a chargeable item which is used for confirmation purpose can be considered as an update followed by a cancel. This means that the chargeable item used for confirmation purpose is processed and leads to the generation of a charged item, aggregated with a pending charged item, and then automatically reported into the external billing system



Multiservice Sessions

A multiservice session can be started, updated or stopped. Then, business sub-operations can be performed inside a multiservice session. The following table lists the different sub-operations which can be performed during a multiservice session depending on whether the session starts, is updated or stops.

A multiservice session can include several sequences of quota reservations. Each sequence is represented by a charging session.


Operation	Sub-Operation(s)
Start	Reserve
Update	Reserve Confirm and reserve Confirm Cancel
Stop	Confirm Cancel


The following table lists the different operations which can be performed on a multiservice session:


Operation	Description
Start	<p>Starting a multiservice session consists in:</p> <ul style="list-style-type: none"> ■ Creating the multiservice session ■ Creating at least one charging session (using a Reserve operation) <p> NOTE</p> <p>A charging session refers to one service but a service can be referred to by several charging sessions.</p> <p>A multiservice session needs the following information to be created:</p> <ul style="list-style-type: none"> ■ A unique external session identifier ■ An SAP CC access, made up with a customer service identifier (SID) and a user service identifier (USID) related to the end customer. The SID must be configured in the charges customized in the charge plans and offers of the service provider. In case of multiservice, this identifier is the same for all the services ■ A start date, which is combined with the above access to retrieve the adequate subscription or provider contract item <p>Optionally, a multiservice session includes:</p> <ul style="list-style-type: none"> ■ The identifier of a reservation renewal listener (to enable the reservation renewal mechanism) ■ External data embedded in the renew reservation notifications which is sent to the listeners in the registered client application (or system) <p> NOTE</p> <p>A reservation renewal listener must initially be registered. This customized listener is implemented in the client application or system. This can be the charging client application or another client application in charge of handling the renew reservation notifications.</p> <p>Then, you can define several charging sessions using multiple Reserve operations.</p>
Update	<p>To update a multiservice session, a client application must provide the following information:</p> <ul style="list-style-type: none"> ■ The unique identifier of the session, which corresponds to the one provided at start time ■ A service identifier (SID) and a user service identifier (USID), which together define an access ■ An update date, which must correspond to the start date in order to match the correct access chronology <p>Then, you can create a new charging session or modify existing charging sessions using the Reserve, Confirm, Confirm and Reserve, or Cancel operations, which gives you the possibility to:</p> <ul style="list-style-type: none"> ■ Create one charging session per Reserve operation ■ Confirm and extend a reservation of an existing charging session with the Confirm and Reserve operation


	<ul style="list-style-type: none"> ■ Confirm a reservation and stop the related charging session with the Confirm and Reserve operation ■ Cancel the last reservation of an existing charging session and stop the charging session
Stop	<p>Stopping a multiservice session consists in:</p> <ul style="list-style-type: none"> ■ Confirming or cancelling reservations of existing charging sessions ■ Deleting the multiservice session ■ Cleaning the remaining charging sessions (using the default resolution)

The following table precisely describes the sub-operations which can be performed when starting, updating, or stopping a multiservice session:

Business Operation	Followed on by	Description
Reserve	Confirm and Reserve Confirm Cancel	<p>Creates a charging session and reserves a given number of service units. The Reserve operation is called with the following information:</p> <ul style="list-style-type: none"> ■ The identifier of the reservation, also used to identify the related charging session (the pair (reservation ID, multiservice session ID) identifies the charging session) ■ A chargeable item, which includes the requested quota of service per property and is used for reservation purposes ■ A default resolution, which determines the behavior of the charging session at expiration time ■ A time-to-live (TTL), which defines the expiration time of the charging session ■ The name of a property of the chargeable item to be inverted and which used by the best-effort reservation mechanism <p>The SAP CC system:</p> <ul style="list-style-type: none"> ■ Simulates the charging operation of the provided chargeable item ■ Performs the adequate quota reservations (or best-effort reservations if service units are not sufficient) ■ Creates a charging session, keeping the original chargeable item used for reservation purpose ■ Returns a rating result which can be used to impact the related account balance <p> NOTES</p> <ul style="list-style-type: none"> ■ When the system performs best-effort reservation, the reservation is partially granted. The charging session is then eligible to the reservation renewal mechanism which will notify the registered listeners when new reservations can be performed due to balance re-credit ■ When the system performs a reservation which is fully granted, the sending of renew reservation notifications for this charging session is deactivated

<p>Confirm and reserve</p>	<p>Confirm and Reserve Confirm Cancel</p>	<p>Gives the number of service units which were consumed and reserves other service units. Note that the number of consumed units can be different from the number of units initially reserved.</p> <p>The Confirm and Reserve operation is called with the following information:</p> <ul style="list-style-type: none"> ■ The identifier of the reservation identifier, also used to identify the related charging session ■ An optional chargeable item used for confirmation purpose, which is based on the chargeable item previously used for reservation. If the chargeable item was not modified (compared to reservation) or if it is not present, the reservation is fully confirmed. If the chargeable item related to this reservation was modified, the confirmation is partial ■ A chargeable item, which includes the requested quota of service per property and is used for reservation purposes ■ A time-to-live (TTL), which defines the expiration time of the charging session <p>First of all, SAP CC retrieves the concerned charging session and definitively charges the reservation chargeable item stored in this session. A rating result is then created and contains:</p> <ul style="list-style-type: none"> ■ Optional transactions (*) used to cancel the previous reservation operation which impacted the related account balance ■ An amount to cancel. The amount to cancel is the sum of the amounts contained in the above-mentioned transactions to cancel <p>Secondly, SAP CC creates a consolidated chargeable item by comparing:</p> <ul style="list-style-type: none"> ■ The reservation chargeable item stored in the charging session ■ The confirmation chargeable item potentially provided (or automatically re-created) <p> NOTES</p> <ul style="list-style-type: none"> ■ The consolidated chargeable item is rated and previously reserved counters are updated in both database and memory ■ The previously created rating result is enhanced and contains: <ul style="list-style-type: none"> ● Optional transactions used to confirm the previous reservation operation which impacted the related account balance ● An amount to confirm. The amount to confirm is the sum of the amounts contained in the above-mentioned transactions to confirm ■ The transactions stemmed from this rating result are sent to the TIF and written into the adequate data files <p>Thirdly, SAP CC creates a new reservation chargeable item by consolidating:</p> <ul style="list-style-type: none"> ■ The reservation chargeable item stored in the session (coming from the previous reservation)
----------------------------	---	---

		<ul style="list-style-type: none"> ■ The reservation chargeable item provided in the Confirm and Reserve operation potentially empty, partially or completely filled <p> NOTES</p> <ul style="list-style-type: none"> ■ This new reservation chargeable item is rated and concerned counters are reserved: <ul style="list-style-type: none"> ■ Systematically for persistent counters ■ Potentially for transient counters according to their configuration (see above for explanation about transient counters update during reservation) ■ The previously created rating result is enhanced and contains: <ul style="list-style-type: none"> ● Optional transactions used to reserve quota for the current reservation ● An amount to reserve. The amount to reserve is the sum of the amounts contained in the above-mentioned transactions to reserve <p>Finally, SAP CC updates the charging session with this new consolidated reservation chargeable item considered as the new reference for the next operation (Confirm, Confirm and Reserve, or Cancel).</p>
Confirm		<p>Gives the number of service units which were really consumed and cancels the related charging session.</p> <p>The Confirm operation is called with the following information:</p> <ul style="list-style-type: none"> ■ The identifier of the reservation identifier also used to identify the related charging session ■ An optional chargeable item used for confirmation, based on the chargeable item previously used for reservation. If the chargeable item was not modified (compared to reservation) or if it is not present, the reservation is fully confirmed. If the chargeable item related to this reservation was modified, the confirmation is partial. <p>First of all, SAP CC retrieves the concerned charging session and charges again the reservation chargeable item stored in this session. A rating result is then created and contains:</p> <ul style="list-style-type: none"> ■ Optional transactions (*) used to cancel the previous reservation operation which impacted the related account balance ■ An amount to cancel. The amount to cancel is the sum of the amounts contained in the above-mentioned transactions to cancel <p>Secondly, SAP CC creates a consolidated chargeable item by comparing:</p> <ul style="list-style-type: none"> ■ The reservation chargeable item stored in the session ■ The confirmation chargeable item potentially provided (or automatically re-created)

		<p> NOTES</p> <ul style="list-style-type: none"> ■ The consolidated chargeable item is rated and previously reserved counters are updated in both database and memory ■ The previously created rating result is enhanced and contains: <ul style="list-style-type: none"> ● An amount to confirm ● Optional transactions used to confirm the previous reservation operation which previously impacted the related account balance ■ The transactions stemmed from this rating result are sent to the TIF and written into the adequate data files <p>Thirdly, SAP CC removes the charging session and the identifier of the last reservation becomes available for other reservations.</p>
Cancel		<p>Cancels the last reservation and removes the related charging session.</p> <p>The Cancel operation is called with the identifier of the reservation which is also used to identify the related charging session</p> <p>SAP CC:</p> <ul style="list-style-type: none"> ■ Retrieves the concerned charging session and cancels counter reservations related to the last reservation ■ Returns an empty rating result ■ Removes the charging session and makes the identifier of the last reservation available for other reservations

(*) The generation of the transactions to cancel is deactivated by default for performance reasons, but can be activated using the `TRANSACTIONS_TO_CANCEL_IN_RESULT_ENABLED` parameter. The generation of the transactions to cancel must be activated if balances are handled by an external system. If the generation of the transactions to cancel is deactivated, the amount to cancel is not computed. For further information about the `TRANSACTIONS_TO_CANCEL_IN_RESULT_ENABLED` parameter refer to the [SAP Convergent Charging Parameters Reference](#) documentation.

Reservation Renewal

The Reservation Renewal mechanism is an optional mechanism of multiservice sessions which gives the possibility for a client application or system to reserve again a quota of service in a given charging session which is inactive due to insufficient credit. This mechanism relies on listeners which are implemented in the client applications or systems, and which listen to specific notifications (renewal reservation notifications) sent by the Raters of the SAP CC system. These notifications are sent to the registered listener when:

- A charging session has been partially granted (final units)
- An event occurred in SAP CC which may release service units (monetary balance or counters)

The Reservation Renewal mechanism informs a registered client application or system that it should renew its current credit reservations because the context of an end customer has changed during a charging session, due to an event which may have released service credits (monetary balance or counters), for example:

- A refill request on one of the customer's prepaid accounts in the subscriber account relating to the end customer
- A parallel charging session which confirms a reservation using values different from the reservation request
- A parallel charging session which is cancelled either by the client application or when a reservation is reaching its "time-to-live" (TTL)

The Reservation Renewal feature is optional and available when a reservation is partially granted during a charging session. The requested credit cannot be fully reserved and final units are reserved.



EXAMPLE

When implementing credit-control features, the renew reservation notifications can be used to implement the management of re-authorization requests (RAR) and re-authorization answers (RAA) defined by the RFC 4006.

Registered Client Application

A charging client application performing session-based charging operations and using multiservice sessions can implement the reservation renewal mechanism, using a customized reservation renewal listener which must be registered to SAP CC in order to receive and handle the renew reservation notifications.

Notification Send and Resend

SAP Convergent Charging sends renewal reservation notifications to the registered client applications or systems. In case of notification failure, SAP CC regularly resends the renew reservation notifications to the registered client applications (or systems) according to the system settings (attempts, intervals).

Process Limitations

The configuration of price plans dedicated to session-based charging operations must comply with the following limitations:

- The rate components which are used within price plans to compute usage rates must ensure an **increasing** computed amount. Rate components which lead to decreasing amounts are forbidden and lead to inconsistency (particularly during the reservation operation)
- As usage rates can depend on counter values, the set or reset counter operations must not be applied on counters which are used during session-based charging operations. This limitation ensures a positive balance for the related prepaid accounts. For further information about the counter operator component, refer to the [SAP Convergent Charging Logic Components Reference](#) documentation



EXAMPLE

The following example demonstrates a counter reset (every 30 min.) which occurs during a session relating to a phone call:

		Database state	Current State	Reservation
Session Start (int. call)	→	Counter: 6 min.	Counter: 6 min.	
Session OK (3 min. granted)	← \$0	<i>Unchanged</i>	Counter: 3 min.	Counter -3
Reset to 30 min	→	Counter: 30 min.	<i>Unchanged</i>	
Session Stop (2 min used)	→	<i>Unchanged</i>	Counter: 33 min.	
Session End	← \$0	Counter: 28 min.	Counter: 28 min.	

Use Cases

The following use cases illustrate the session-based charging process and particularly the reservation mechanism.

Use case 1: Single session with a 3 minutes reservation quota on a prepaid account which has 4 available minutes of communication. The following price plan is used:

- \$ 0.40 / minute for international calls
- \$ 0.20 / minute for national calls

Session operation		Cost	Database	Memory	Reservations
Start (int. call)	→		C=4	C=4	
OK (3 min. granted)	←	\$0	Unchanged	C=1	C-3
Stop (2½ min used)	→		Unchanged	C=4	
End	←	\$0	C=1.5	C=1.5	

C: Counter of available minutes

- **Session Start:** Database and memory counters correspond to the 4 minutes available for international or national calls. 3 minutes are reserved and the memory counter value is thus set to $4-3=1$. As this value is positive, no specific cost is computed yet
- **Session Stop:** The previous reservation is cancelled, which thus modifies the memory counter value and sets it back to $1-(-3)=4$. The 2½ consumed minutes are decreased from the database and memory counters, which gives 1½ minute of communication available for future sessions

Use case 2: Two sessions with a 3 minutes reservation quota on a prepaid account which has 4 available minutes of communication. The following price plan is used:

- \$ 0.40 / minute for international calls
- \$ 0.20 / minute for national calls

Session operation		Cost	Database	Memory	Reservations
S1 Start (int. call)	→		C=4	C=4	
S1 OK (3 min. granted)	←	\$0	Unchanged	C=1	C-3
S2 Start (nat. call)	→		Unchanged	Unchanged	
S2 OK (3 min. granted)	←	\$0.4	Unchanged	C=0	C-1 (...C-3)
S2 Stop (2 min used)	→		Unchanged	C=1	C-3
S2 End	←	\$0.2	C=3	C=0	
S1 Stop (2½ min used)	→		Unchanged	C=3	
S1 End	←	\$0	C=0.5	C=0.5	

Sx: Session X, C: Counter of available minutes

- **Session1 Start:** Database and memory counters correspond to the 4 minutes available for international or national calls. 3 minutes are reserved and the memory counter value is thus set to $4-3=1$. As this value is positive, no specific cost is computed yet
- **Session2 Start:** 3 minutes should be reserved but the memory counter indicates that only 1 minute is still available. This minute is thus reserved, and decreased from the memory counter whose value is thus set to $1-1=0$. The 2 other required minutes are rated and correspond to a \$0.4 consumption price

- ➔ **Session2 Stop:** The previous reservation (of 1 minute) is cancelled, which thus modifies the memory counter value and sets it back to $0 - (-1) = 1$. 1 minute of the 2 consumed minutes is rated (and corresponds to a \$0.2 consumption price), and the other consumed minute is decreased from the database and memory counters, which are respectively updated to 3 and 0 minutes available for future sessions
- ➔ **Session1 Stop:** The previous reservation (of 3 minutes) is cancelled, which thus modifies the memory counter value and sets it back to $0 - (-3) = 3$. The $2\frac{1}{2}$ consumed minutes are decreased from the database and memory counters, which gives $\frac{1}{2}$ minute of communication available for future sessions. As this value is positive, no additional cost is computed

 **NOTE**

When session2 ends, the memory counter value is set to 0 available minutes. This means that no new reservation can be done for this session.

Use case 3: Two sessions with a 6 minutes reservation quota on a prepaid account which has 20 available minutes of communication. The following price plan is used:

- \$ 0.40 / minute for international calls
- \$ 0.20 / minute for national calls

Session operation		Cost	Database	Memory	Reservations
S1 Start (int. call)	➔		C=20	C=20	
S1 OK (6 min. granted)	←	\$0	Unchanged	C=14	C-6
S2 Start (nat. call)	➔		Unchanged	Unchanged	
S2 OK (6 min. granted)	←	\$0	Unchanged	C=8	C-6 (...C-6)
S2 Stop (2 min used)	➔		Unchanged	C=14	C-6
S2 End	←	\$0	C=18	C=12	
S1 Stop ($2\frac{1}{2}$ min used)	➔		Unchanged	C=18	
S1 End	←	\$0	C=15.5	C=15.5	

Sx: Session X , C: Counter of available minutes

- ➔ **Session1 Start:** Database and memory counters correspond to the 20 minutes available for international or national calls. 6 minutes are reserved and the memory counter value is thus set to $20 - 6 = 14$. As this value is positive, no specific cost is computed
- ➔ **Session2 Start:** The memory counter indicates that 14 minutes are potentially available, and that 6 minutes can thus be reserved for this national call. The memory counter is updated to $14 - 6 = 8$. As this value is positive, no specific cost is computed
- ➔ **Session2 Stop:** The previous reservation (of 6 minutes) is cancelled, which thus modifies the memory counter value and sets it back to $8 - (-6) = 14$. The 2 consumed minutes are decreased from the database and memory counters, which are respectively updated to $20 - 2 = 18$ and $24 - 2 = 12$ minutes available for future sessions. As this value is positive, no additional cost is computed
- ➔ **Session1 Stop:** The previous reservation (of 6 minutes) is cancelled, which thus modifies the memory counter value and sets it back to $12 - (-6) = 18$. The $2\frac{1}{2}$ consumed minutes are decreased from the database and memory counters, which gives $15\frac{1}{2}$ minutes of communication available for future sessions. As this value is positive, no additional cost is computed

Use case 4: Single session (dealing with session update) with a 6 minutes reservation quota on a prepaid account which has 30 available minutes of communication. The following price plan is used:

- \$ 0.40 / minute for international calls
- \$ 0.20 / minute for national calls

Session operation		Cost	Database	Memory	Reservations
Start (int. call)	→		C=30	C=30	
OK (6 min. granted)	←	\$0	Unchanged	C=24	C-6
Update (2 min used)	→		Unchanged	C=30	
OK (6 min. granted)	←	\$0	C=28	C=22	C-6
Stop (2 min used)	→		Unchanged	C=28	
End	←	\$0	C=26	C=26	

C: Counter of available minutes

- ➔ Session **Start**: Database and memory counters correspond to the 30 minutes available for international or national calls. 6 minutes are reserved and the memory counter value is thus set to $30-6=24$. As this value is positive, no specific cost is computed
- ➔ Session **Update**: The previous Reservation (of 6 minutes) is cancelled, which thus modifies the memory counter value and sets it back to $24-(-6)=30$. The database counter is decreased from the 2 consumed minutes, and the memory counters is decreased from the 2 consumed minutes and the new Reservation (of 6 minutes). These counters are thus respectively updated to $30-2=28$ and $30-2-6=22$ minutes available for future sessions
- ➔ Session **Stop**: The previous reservation (of 6 minutes) is cancelled, which thus modifies the memory counter value and sets it back to $22-(-6)=28$. The 2 consumed minutes are decreased from the database and memory counters, which gives 26 minute of communication available for future sessions. As this value is positive, no additional cost is computed



NOTE

The behavior during the session update operation is equivalent to a session stop operation followed by a session start operation.

Use case 5: Single session (dealing with session update) with a 6 minutes reservation quota on a prepaid account which has 1 available minute of communication and a \$2.8 initial balance. The following price plan is used:

- \$ 0.40 / minute for international calls
- \$ 0.20 / minute for national calls

Session operation		Cost	Database	Memory	Reservations
Start (int. call)	→		C=1 b=2.8	C=1 b=2.8	
OK (6 min. granted)	←	\$2	Unchanged Unchanged	C=0 b=0.8	C-1 b-2
Update	Stop (2 min used)	→	Unchanged Unchanged	C=1 b=2.8	
	OK	←	C=0 b=2.4	C=0 b=2.4	

Start (6 min required)	→		Unchanged Unchanged	Unchanged Unchanged	
OK	←	\$2.4	Unchanged Unchanged	Unchanged b=0	b-2.4
Stop (2 min used)	→		Unchanged Unchanged	Unchanged b=2.4	
End	←	\$0.8	C=0 b=1.6	C=0 b=1.6	

C: Counter of available minutes, b: Balance of the prepaid account

- **Session Start:** Database and memory counters correspond to the 1 minute available for international or national calls. 6 minutes should be reserved but the memory counter indicates that only 1 minute is still available. This minute is thus reserved, and decreased from the memory counter whose value is thus set to $1-1=0$. The 5 other required minutes are rated and correspond to a \$2 consumption price. These \$2 are reserved, and the balance stored in memory is updated to $2.8-2=0.8$
- **Session Update:** As explained above, a session update is equivalent to a session stop followed by a session start.
 - **Stop operation:** Previous counter (1 minute) and balance (\$2) reservations stored in memory are cancelled, which respectively sets back their values to $0-(-1)=1$ and $0.8-(-2)=2.8$. Then, the 2 consumed minutes are decreased from counter and balance values (database and memory values), which are thus respectively updated to $1-1=0$ and $2.8-0.4=2.4$. A \$0.4 cost corresponding to 1 minute of international call is returned
 - **Start operation:** A 6 minutes reservation must be performed, but the memory counter indicates that 0 minute is available. These 6 minutes are thus rated and the memory balance is decreased from \$2.4, and is thus set to $2.4-2.4=0$. A \$2.4 cost corresponding to 1 minute of international call is returned
- **Session Stop:** The previous \$2.4 Reservation is cancelled, the memory counter is thus set back to $0-(-2.4)=2.4$. The 2 extra consumed minutes corresponding to a \$0.8 cost are decreased from the database and memory balances, which finally give 0 minute of communication available for future sessions and a \$1.6 final balance.

NOTE

When the available prepaid balance is not sufficient to make the reservation of N minutes, an inverse rating operation is performed to compute the maximum number of minutes which can be reserved. This new value prevails over the defined reservation.

Chargeable Items Acquisition Process

Process Overview

The Chargeable Items Acquisition process is used to record incoming chargeable items produced by a client application or system (such as the IEC, a mediation system and so on) into a data storage such as the BART database or a third-party system such as SAP Convergent Invoicing (SAP CI) in the SAP ERP system.

SAP CC gives the possibility to acquire chargeable items:

- Using the BART Server system in a synchronous mode
- Using the Core Server system in an asynchronous mode

For functional information about this feature, refer to the description of the [Chargeable Items Acquisition](#) feature.

Execution Modes

Acquisition using the BART Server

The execution of the Chargeable Items Acquisition process can be performed by the BART Server system, relying on multiple acquisition sessions. These sessions are:

- Usually created by a mediation system using dedicated APIs provided by the BART Server system
- Stored into the BART database to keep track of:
 - The number of newly acquired CDRs
 - The number of duplicated CDRs
 - The number of consolidated CDRs (when consolidation is performed)
 - The elapsed time of the session

To start a new acquisition session, it is necessary to specify some specific parameters:

- The acquisition mode, used to change the behavior of the acquisition process:
 - **ACQUIRE**, used to acquire all CDRs without trying to detect duplicated items
 - **ACQUIRE_AND_DEDUPLICATE**, used to acquire CDRs and try to detect duplicated items. When duplicated items are detected, they are stored in the BART database with a “DUPLICATE” status
 - **ACQUIRE_AND_REJECT_DUPLICATE**, used to acquire CDRs and try to detect duplicated items. When duplicated items are detected, they are rejected and thus not stored in the BART database
- The CDR source, used to specify the origin of the CDRs which must be acquired
- A description of the acquisition session, for information purposes
- The communication protocol which must be used to communicate:
 - XML over HTTP, used to transport proprietary XML messages and commonly used for all API operations
 - Packets over TCP/IP, used to faster transport proprietary packets and thus highly recommended for performing acquisition sessions



NOTES

- The detection of duplicated CDRs relies on comparison between 16-byte MD5 hash values containing selected CDRs fields and called “CDR magic numbers”. A FIFO data structure is allocated to every acquisition session and contains the last 5000 magic numbers. The duplicated items detection thus consists in looking into this FIFO structure every time a CDR is acquired (and thus every time a new magic number is computed). For further information about the magic numbers configuration, refer to the [SAP Convergent Charging Configuration Guide](#) documentation
- For further information about the communication channels, refer to the [SAP Convergent Charging Security Guide](#) documentation

Every started acquisition session has an internal status which gives the possibility to inform about the lifecycle of the session:

- **IN_PROGRESS**, which is used for running sessions
- **ENDED**, which means that the acquisition session is completed without any errors
- **ENDED_WITH_ERROR**, which means that the acquisition session completed with some errors. When the provided BART Server API is used, the mediation chain keeps track of errors
- **CLEANED**, which is used to inform the acquisition session was cleaned on the BART server startup. This status can occur when the BART Server is shut down while an acquisition session is running

Acquisition using the Core Server

The Core Server system acquires chargeable items from the IEC or client applications when the Core Server is configured to work in conjunction with SAP CI (with the consumption item management function enabled). This process uses the following communication channels:

- Packets over TCP/IP, to acquire the incoming chargeable items
- RFC over TCP/IP, to send consumptions items to SAP CI

During the acquisition process, the Core Server system:

- Determines the list of provider contracts which are associated to the chargeable items
- Enriches the chargeable items with additional information
- Validates these enriched chargeable items
- Converts these enriched chargeable items into consumption and sends them to SAP CI for billing purposes (refer to the schema in the [Chargeable Items Acquisition](#) section of this documentation)

An acquisition operation performed by the Core Server system is made up with the following steps (refer to the schema in the [Chargeable Items Acquisition](#) section to get an overview of this execution):

1. A mediation system sends a list of chargeable Items to the Core Server which acquires each chargeable item with access information and consumption date
2. The Dispatcher associates each chargeable item to the relating charging contract or subscription. The Dispatcher then sends acquisition requests to the Raters which build a charged transaction set for each chargeable item. The context of each charged transaction set is filled with the appropriate data, i.e. subscriber account identifier, contract identifier, and so on
3. The Rater processes the charged transaction sets by using the Chargeable Item File processor in order to:
 - Convert these charged transaction sets into chargeable items (refer to the description of the [Chargeable Items Charging Process](#) for further information about this processor)
 - Check and validate that the chargeable items match the related consumption item mapping
 - Write the chargeable items in CSV data files
4. The Bulkloader asynchronously reads the CSV data files and converts every loaded chargeable item into a consumption item
5. The Bulkloader finally sends these consumption items to SAP CI



CAUTION

The mediation system must not send chargeable items with empty string properties. Any specific string like "UNDEF" can be used instead of empty strings.



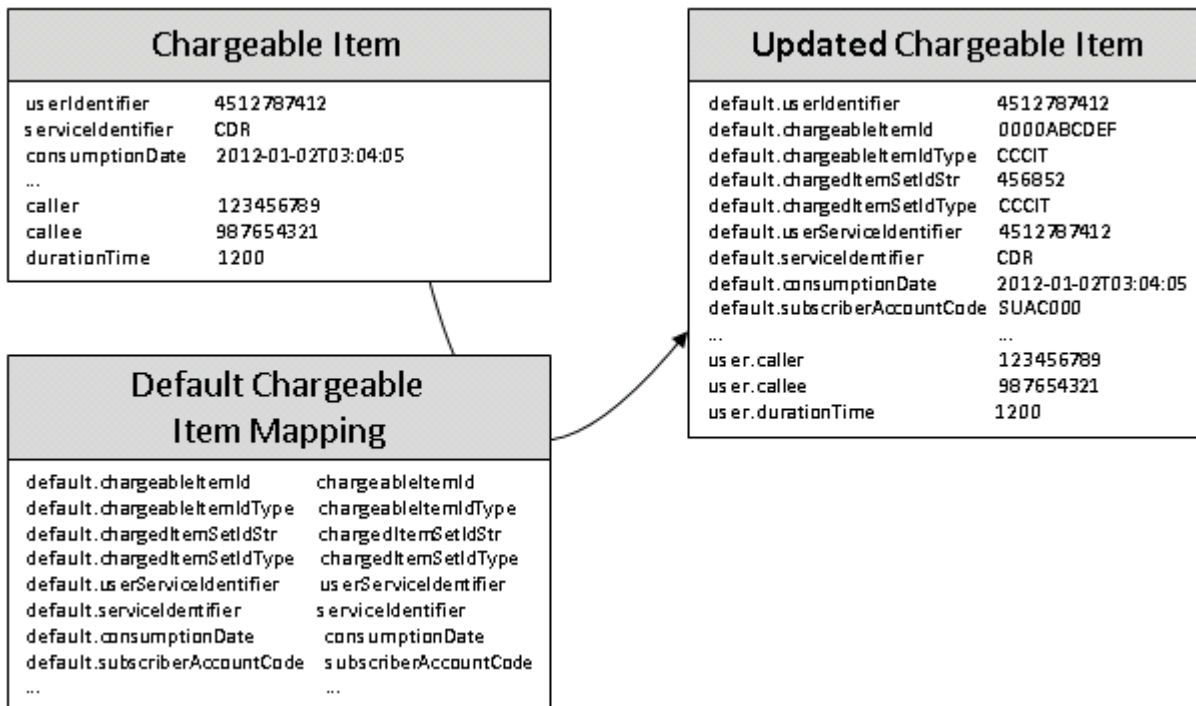
NOTES

- A user granted with the administrator role can export the configuration file of the CIF from the Core database in order to modify it (refer to [Bulk Loading Mechanism and Error Management](#) section for further information)
- The Chargeable Item File Processor executes the following operations to convert charged transaction sets into chargeable items:
 - It retrieves the chargeable item class of the input chargeable item
 - It retrieves the default chargeable item mapping which must be used to export chargeable items (using the "CC" external system code)
 - It retrieves each property of the input chargeable item, adds it to the output chargeable item, and prefixes it with the "user." string

- For each field of the chargeable item mapping, it adds a property (whose name corresponds to the name of the mapping field) to the output chargeable item

EXAMPLE

The following schema illustrates the use of a default chargeable item mapping, which gives the possibility to update the properties of a chargeable item (coming from the charging context) with specific default properties:



Chargeable Items Consolidation Process

Process Description

The Chargeable Items Consolidation process is used to retrieve complementary information related to the acquired chargeable items. This process represents an internal step of the Chargeable Items Acquisition process described above, and can be automatically or manually (through bulk operations) executed.

This section details the execution mode based on BART Server.

Consolidation Operations

To rate the acquired CDRs, the BART Server system needs to interact with the Core Server system in order to retrieve the following additional information for each CDR:

- The identifier of the related provider contract or subscription
- The batch rating group of provider contract or subscription

This additional information is stored into the BART database for persistency purpose. It is also stored into a temporary consolidation cache:

- Handled by the BART Server in order to guarantee performances
- Associated to a given acquisition session and thus deleted at the end of the related session



NOTES

- The “from” date and “to” date properties are also stored in this cache to determine a validity period for the cached data
- These cached data represent about 100-150 bytes per cached access (depending on the length of user and service identifiers plus the lengths of access chronology)
- The BART Server cannot consolidate all the types of subscriptions. It can only consolidate CDRs related to **hybrid** and **offline** subscriptions. In case of consolidation failure, the status of the consolidated CDR is updated in the BART database with the following values:
 - “NO_PROVISIONING”, to inform that the CDR has not been consolidated
 - “ONLINE_SUBSCRIPTION”, to inform that the CDR refers to an online subscription

Tax Calculation Process

Process Overview

For functional information about this feature, refer to the description of the [Tax Calculation](#) feature. The tax calculation process groups two distinct functions:

- Preliminary tax computing, which concerns rated transactions
- Tax adjusting and applying, which concerns charged transactions

The overall result is tax data recorded in all “charged & taxed” transactions specifying the total amount which must be debited on a customer account (postpaid or prepaid):

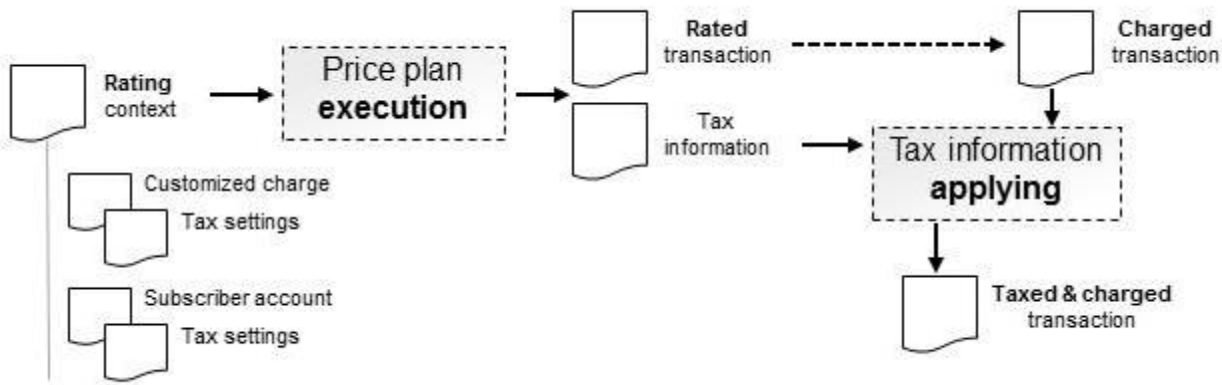
- SAP CC debits the balance of the identified prepaid account with the total amount including taxes
- An external billing and invoicing system consolidates tax data relating to different charging operations in order to calculate the final tax



NOTE

In an integrated scenario with the SAP ERP/FI-CA component of the SAP Business Suite, SAP CC generates and posts valid billable items with tax data.

The following schema provides an overview of the process execution:



Inputs

Input	Description	
Content of a rated transaction	<ul style="list-style-type: none"> ■ Consumption date ■ Price plan amount computed by SAP CC 	
Analyzed Event SAP CC System	Taxation data configured for a tax system	
Master Data (End Customer)	Tax settings of the customer or account holder as set in the subscriber account	Pricing settings that identify a customized charge in the pricing catalog of the service provider
Master Data (Pricing Catalog)	Tax settings for the customer service and for the service provider as set in the charges customized in the charge plan (or in the offer subscribed to) . These customized charges specify a price plan and a charging plan that are executed by SAP CC	

Generated Tax Data

In Rated Transactions

The rated transaction includes:


- A rated tax amount, which corresponds to the tax amount computed by the tax computing function in SAP CC
- A rated amount, which corresponds to the price amount computed by the execution of the price plan and before computing the tax. When the tax system is EU VAT or flat tax, this amount may be net or gross depending on the tax configuration in the charge plan (or in the offer)
- A rated total amount, which corresponds to the total price amount of the transaction, and which always includes the tax

In Charged and Taxed Transactions

The “charged & taxed” transactions are converted to charged items and billable items. They include:

- Tax data
 - Amount (Excl. Tax)
 - Tax Amount
 - Total Amount (Incl. Tax)
 - Tax Code

- Tax Status
- Tax data details

Output Data	Technical ID	Description
Amount (Excluding Tax)	amount	The net amount computed by SAP CC, which can be: <ul style="list-style-type: none"> ■ The base amount calculated during the execution of the price plan ■ The base amount less the tax
Tax Amount	taxAmount	The amount of tax calculated during the execution of the price plan and which is applied to one or several charged transactions
Total Amount (Including Tax)	totalAmount	The computed gross amount which is debited from the balance of the prepaid accounts, and which can be: <ul style="list-style-type: none"> ■ The base amount calculated during the execution of the price plan ■ The base amount less the tax
Tax Code	taxCode	The tax code of a charged transaction (or charged item) assigned to a prepaid account or external postpaid account. The format depends on the invoicing tax system applied during the tax calculation: <ul style="list-style-type: none"> ■ NO_TAX ■ /<COUNTRY_CODE>/<VAT_RATE_LEVEL>#<TIMESTAMP> ■ /RAW/<FLAT_TAX_RATE_ID> ■ EZTAXED <div style="border-left: 2px solid #0070C0; padding-left: 10px; margin-top: 10px;">  NOTE For further information about tax codes, please refer afterwards. </div>
Tax Status	taxStatusCode	The tax status corresponds to the final execution status of the tax calculation process. It specifies if the calculated tax amount has been applied to the total amount or not, using the following possible values. <ul style="list-style-type: none"> ■ NO_TAX, used to indicate that no tax was applied ■ FOR_INFO, used to indicate that the computed tax data is provided for information purpose ■ APPLIED, used to indicate that the tax was applied ■ TAX_EXEMPTED, used in case of tax exemption ■ BUYER_SUBJECT_TO_PAY, used to indicate that the buyer is in charge of reversing the tax to the tax authority When the tax is not applied to the total amount, the tax status indicates the reasons for the following situations: <ul style="list-style-type: none"> ■ Tax amount is calculated for information purpose (EU VAT) ■ Customer is subjected to pay the tax in his country ■ Customer is exempted to pay the tax



For a complete list of generated fields related to tax calculation, refer to the [SAP Convergent Charging Configuration Guide](#) documentation.

In Billable Items

In an integrated *SAP Solution* scenario with the SAP ERP/FI-CA component of the SAP Business Suite, SAP CC generates valid billable items.


Tax data computed by SAP CC	Technical ID	Fields in sent BITS (*Property in Charged Items)
<p>Base Amount</p> <p>SAP CC generates this output data with the amount calculated during the execution of the price plan. This amount can be net or gross depending on the configuration of the charge customized in a charge plan (or in an offer)</p>	baseAmount	<p>BIT_AMOUNT</p> <p><i>Price amount computed by SAP CC</i></p>
VAT Transaction Gross Price Flag	taxVatGrossPriceFlag	<p>TAX_INCLUDED</p> <p><i>SAP ERP determines if the computed price (BIT_AMOUNT) is net or gross.</i></p>
<p>Tax Determination Type</p> <p>SAP CC generates this output data depending on the tax system used during the tax calculation process. Possible values are:</p> <ul style="list-style-type: none"> ■ "01": VAT or Flat ■ "04": US Telco Taxes (EZTax) 	taxDetType	<p>TAX_DET_TYPE</p> <p><i>SAP ERP expects 01 for VAT and 04 for EZTax.</i></p>
SAP CC does not generate this output data. You must set up the generation of charged items in the relevant charge plan or in the relevant charged item class	N/A	EXT_TAX_DATE
VAT Transaction Rate Code	taxVatRateCode	<p>EXT_TAX_ID</p> <p><i>SAP ERP receives the determined Tax Information as external tax code (EXT_TAX_ID) to be mapped in posting area 8123.</i></p>
VAT Place of Taxation	taxVatTaxationPlace	TAX_COUNTRY

NOTE

SAP CC generates the EXT_TAX_ID and TAX_COUNTRY fields depending on the VAT rules selected in the charges customized in a charge plan activated in a provider contract.

Tax Code

The format of the tax code depends on the invoicing tax system which is finally used during the tax computation.

Tax System	Syntax and Description
Value Added Tax (VAT) System	<p>Syntax: <code>/<Country_Code>/<VAT_Rate_Level>#<Timestamp></code></p> <p>Where:</p> <ul style="list-style-type: none"> ■ <code>Country_Code</code> is the ISO 3166 country code (2-digit) ■ <code>VAT_Rate_Level</code> is a value between 0 and 5 ■ <code>Timestamp</code> corresponds to the number of days between January 1st, 1970 and the validity start date of the tax rate value <p> EXAMPLE</p> <p>The rate value for the normal tax rate level is 0.196 since 05/01/01. A charged item can include the following tax code: <code>/FR/0#11443</code></p>
Flat Tax System	<p>Syntax: <code>/RAW/<Rate_Identifier></code></p> <p>Where <code>Rate_Identifier</code> corresponds to the identifier of the flat tax rate set in the charge customized in a charge plan (or in an offer)</p>
EZTax System	<p>Syntax: EZTAXED</p>
Special	<p>Syntax: NO_TAX</p> <p>The following reasons explain this output value:</p> <ul style="list-style-type: none"> ■ No tax can be set in VAT rate level of the charge customized in a charge plan ■ No tax can be set in the charge customized in a charge plan ■ No tax can be set in the subscriber account ■ No tax can be set by SAP CC when a configuration is erroneous: The charge plans and subscriber accounts are not compliant with <p>For more information, refer to the detailed functions in this section.</p>

**NOTE**

You can use the tax code to determine the invoicing tax system used by the SAP CC system to compute the tax data.

Process Execution

Tax Computing

This function starts when SAP CC has dynamically determined the price amount of the processed event during the execution of a price plan. The different rating contexts updated by the Raters contain all the tax information both related to the subscriber account (customer master data) and to the customized charge (pricing catalog master data) associated to the analyzed events.

During the execution of price plans, the Raters create rated transactions and their associated tax information. This tax information contains all the following data which are required by the tax systems to compute the adequate taxes and rated transaction amounts:

- Tax settings of the customer or account holder as set in the subscriber account
- Tax settings for the customer service and for the service provider as set in the charges customized in the charge plan or the offer)
- Taxation data configured for a tax system

The tax computing function takes into account the fact that the prices defined in the price plan of the customized charge are net or gross:

Prices	Computation description	Important note
Net	<ul style="list-style-type: none"> ■ SAP CC considers that the price amount determined by the execution of the price plan does not include the tax ■ The rated amount is the price amount determined during the execution of the price plan ■ SAP CC computes the tax, and then computes the total rated amount 	
Gross	<ul style="list-style-type: none"> ■ SAP CC considers that the price amount determined by the execution of the price plan does not include the tax ■ The total rated amount is the price amount determined during the execution of the price plan ■ SAP CC computes the tax, and then computes the rated amount 	Not compliant with the following taxation modes: <ul style="list-style-type: none"> ■ Account holder subject to pay ■ Account holder exempted



NOTE

When the BillSoft EZTax tax system is selected in a charge customized in a charge plan (or in an offer), SAP CC considers that the prices defined in the price plan of this charge is net. The tax is calculated and added to the total rated amount.

SAP CC computes the rated tax amount depending on the tax system:

Tax system	Net prices	Gross prices
EU VAT, Flat	Tax Rate * Price Plan Amount The tax amount is computed by multiplying a tax rate by price amount determined during the execution of the price plan. SAP CC determines the value of this tax rate depending on the tax system and on the master data.	$\text{Tax Rate} * \text{Price Plan Amount} / (1 + \text{Tax Rate})$ The tax amount is computed using the above formula.
US Telco (EZTax)	Global Tax Amount The tax amount is retrieved from the BillSoft EZTax system.	N/A

SAP CC computes the remaining tax data with the following algorithms:

Computed Tax Data	Net Prices	Gross Prices
Rated Amount (Before Tax Computing)	Price Plan Amount The rated amount is the price amount determined during the execution of the price plan.	Price Plan Amount The rated amount is the sum of the tax amount and the price amount determined during the execution of the price plan. Note that SAP CC does not compute this data by removing the tax amount to the price amount determined during the execution of the price plan (Price Plan Amount – Rated Tax Amount).
Total Rated Amount (Including Tax)	Rated Amount + Rated Tax Amount The total rated amount is the sum of the tax amount and the price amount determined during the execution of the price plan.	Price Plan Amount The total rated amount is the price amount determined during the execution of the price plan.

 **NOTE**

The total rated amount always includes the tax.

Tax Adjusting and Applying

This function starts when SAP CC has split a rated transaction into one or several charged transactions during the execution of the charging plan. The rated amount in the rated transaction is also split into these charged transactions as a base amount. The sum of the base amounts is equal to rated amount. Each charged transaction is assigned to an account (prepaid or external) in the subscriber account previously identified.

The computed taxes are then adjusted. SAP CC determines who is subject to pay the tax and the corresponding tax policy.

Adjusted Tax Data in Charged Transactions	Tax System	Net Prices	Gross Prices
Tax Amount	EU VAT, Flat	Base Amount * Tax Rate SAP CC computes the tax amount by multiplying the tax rate by the base amount. The tax rate is determined during the preliminary tax computing.	Base Amount * Tax Rate / (1 + Tax Rate) SAP CC computes the tax amount using the above formula.
	U.S. Telco (EZTax)	Base Amount * Rated Tax Amount / Total Rated Amount	N/A

Then, SAP CC sets up the amounts (including tax and excluding tax) of the charged transaction.

Adjusted Tax Data in Charged Transactions	Net Prices	Gross Prices
Amount (Including Tax)	Base Amount	Base Amount - Tax Amount
Total Amount (Excluding Tax)	Base Amount + Tax Amount	Base Amount

 **NOTE**

When a rated transaction is split to only one charged transaction, the base amount is equal to the rated amount.

Finally, SAP CC applies (or not) the tax on these charged transactions, which thus become taxed & charged transactions, ready for persistency purposes. It uses the taxation mode configured in the subscriber account to decide how to apply the tax.

Taxation Modes

The following table contains the different possible taxation modes:

Taxation Mode	Tax Adjusting	Tax Applying	
		Postpaid environment	Prepaid environment
Service Provider Subject to Pay	Tax is adjusted and added to the total amount	Adjusted tax is not applied	Adjusted tax is applied
Account Holder Subject to Pay	Tax is adjusted but not applied. The account holder must pay the tax to the tax authority of his residence.	Adjusted tax is not applied	(*) Not applicable
Account Holder Exempted	Tax is adjusted but not applied	Adjusted tax is not applied	(*) Not applicable
None	Tax is not computed	No tax	No tax

(*) Not applicable: SAP CC considers that both the Account Holder Subject to Pay and the Account Holder Exempted taxation modes are not possible in a prepaid environment. The system does not compute the tax and terminates the processing.

Service Provider Subject to Pay

When the taxation mode is set to “Service Provider Subject to Pay”, tax is adjusted and added to the total amount. The applying of this adjusted tax depends on the account type:

- For postpaid accounts, adjusted tax is not applied, which means that an external system must thus manage or consolidate the tax data
- For prepaid accounts, adjusted tax is applied

SAP CC generates the following fields in charged transactions and charged items:

Generated Tax Data	Postpaid charged transaction	Prepaid charged transaction
Amount	Adjusted Rated Amount	Adjusted Rated Amount
Tax Amount	Adjusted Tax Amount	Adjusted Tax Amount
Total Amount	Adjusted Rated Amount + Adjusted Tax Amount	Adjusted Rated Amount + Adjusted Tax Amount
Tax Code	(*)	(*)
Tax Status	FOR_INFO	APPLIED

(*) Depends on the tax system configured to compute the tax.

Account Holder Subject to Pay

When the taxation mode is set to “Account Holder Subject to Pay”, the account holder must pay the tax to the tax authority of his residence. The applying and adjusting of this tax depends on the account type:

- For postpaid accounts, tax is adjusted but not applied, as the account holder must not pay the tax to the service provider but directly to its taxing authority
- For prepaid accounts, tax adjusting and applying is not available. As a consequence, do not use the value computed by the system. It is subject to change

Generated Tax Data	Postpaid charged transaction	Prepaid charged transaction
Amount	Adjusted Rated Amount	Adjusted Rated Amount
Tax Amount	Adjusted Tax Amount	0
Total Amount	Adjusted Rated Amount	Adjusted Rated Amount
Tax Code	(*)	NO_TAX
Tax Status	BUYER_SUBJECT_TO_PAY (Account Holder Subject to Pay)	No tax (degraded function)

(*) Depends on the tax system configured to compute the tax.

Account Holder Exempted

When the taxation mode is set to “Account Holder Subject to Pay”, the account holder does not pay the tax. The applying and adjusting of this tax depends on the account type:

- For postpaid accounts, tax is adjusted but not applied
- For prepaid accounts, tax adjusting and applying is not available.

Generated Tax Data	Postpaid charged transaction	Prepaid charged transaction
Amount	Adjusted Rated Amount	Adjusted Rated Amount
Tax Amount	Adjusted Tax Amount	0
Total Amount	Adjusted Rated Amount	Adjusted Rated Amount
Tax Code	(*)	NO_TAX
Tax Status	TAX_EXEMPTED (Account Holder Exempted)	No tax (degraded function)

(*) Depends on the tax system configured to compute the tax.

No taxation

When the taxation mode is set to “No Taxation”, no tax is computed.

Generated Tax Data	Postpaid charged transaction	Prepaid charged transaction
Amount	Adjusted Rated Amount	Adjusted Rated Amount
Tax Amount	0	0
Total Amount	Adjusted Rated Amount	Adjusted Rated Amount
Tax Code	NO_TAX	NO_TAX

Generated Tax Data	Postpaid charged transaction	Prepaid charged transaction
Tax Status	No tax	No tax

Process Functions

EU VAT with EU VAT Rules

SAP CC uses the EU VAT system depending on the tax settings in the subscriber account or in the customized charge. When these settings are not compliant with this tax system and settings configured in the customized charge, SAP CC does not calculate the tax.

No tax is managed when:

- EU VAT is not selected in the subscriber account
- No VAT rule is specified in the customized charge

Inputs

To compute and adjust value added taxes (VAT), SAP CC needs the following information:

- Information coming from the analyzed event, containing the consumption date
- Information coming from the rating operation (dynamic pricing), containing the price plan amount previously rated by SAP CC
- Information configured in the customized charge, which contains the following data:
 - A flag which defines if the base amount generated during the execution of the price plan are assumed to be net amounts or gross amounts
 - A tax rate level (normal, reduced, super reduced, zero tax, and so on)
 - A VAT rule, which determines the entity that must pay the tax amount and thus impact the tax amounts
 - The country of the service provider (ISO 3166 country code)
- Tax information, which is configured in the subscriber account and contains the following data:
 - The country of the customer
 - The business category (B2B, B2C)
 - The tax system and zone information
 - The taxation mode, which determines the entity that must reverse or not the tax to the tax authority and thus impact the tax amounts according to the following rules:

Taxation mode	Impact
Service Provider Subject to Pay	Tax is calculated and added to the net amount
Account Holder Subject to Pay	Tax is calculated but not applied. The account holder must pay the tax to the tax authority of his residence.
Account Holder Exempted	Tax is calculated but not applied
None	Tax is not calculated

Tax Computing

To compute the tax data, SAP CC first computes the value of the tax rate:

- SAP CC determines the taxation place (country) defined by the VAT rule for the business category (B2B, B2C):

Business Category	Possible Taxation Places
B2B Services: The customer, holding the subscriber account, is a company	<ul style="list-style-type: none"> ■ The customer's place of establishment (default value) ■ The service provider's place of establishment ■ A specific place provided by a property in the rating context (e.g. the consumption place, the delivery place)
B2C Services: The customer, holding the subscriber account, is a private individual	<ul style="list-style-type: none"> ■ The customer's place of residence ■ The service provider's place of establishment (default value) ■ A specific place provided by a property in the rating context (e.g. the consumption place, the delivery place)

- If the VAT rule cannot apply, the default B2B/B2C taxation places set in the VAT rule is applied according to the business category (B2B or B2C). SAP CC selects the country of this place and its country tax policy.
- SAP CC retrieves the country tax policy for the selected country, and determines the value of the tax rate according to the tax level, selected country tax policy, and consumption date.

**NOTE**

For more information about the preconfigured VAT rules and the associated customizing activities, refer to the [Features](#) section.

Then, the Raters compute the tax data included in the rated transactions:

Computed Tax Data in a Rated Transaction	Net Prices	Gross Prices
Tax Rate	SAP CC computes the tax rate value according to: <ul style="list-style-type: none"> ■ The tax rate level and the consumption date ■ The country tax policy identified by the VAT rule 	
Rated Tax Amount	Calculates the tax amount based on the price amount computed during the execution of the price plan, and the tax rate: Tax Rate * Price Plan Amount	$\text{Tax Rate} * \text{Price Plan Amount} / (1 + \text{Tax Rate})$
Rated Amount (Before Tax Computing)	Price Plan Amount	Price Plan Amount
Total Rated Amount (Including Tax)	Rated Amount + Rated Tax Amount	Price Plan Amount

Tax Adjusting and Applying

This function starts when SAP CC has split a rated transaction into one or several charged transactions during the execution of the charging plan. The rated amount in the rated transaction is also split into these charged transactions as a base amount. The sum of the base amounts equals the rated amount. Each charged transaction is assigned to an account (prepaid or external) in the subscriber account previously identified.

The tax rate of the rated transaction is computed.

SAP CC adjusts the computed taxes. It determines the tax amount in the charged transaction depending on the base amount and the tax rate value:

Adjusted Tax Data in Charged Transactions	Net Prices	Gross Prices
Tax Amount	Base Amount * Tax Rate	Base Amount * Tax Rate / (1 + Tax Rate)

Finally, SAP CC applies (or not) the tax on these charged transactions, which thus become taxed & charged transactions, ready for persistency purposes. It uses the taxation mode configured in the subscriber account to decide how to apply the tax. Depending on the taxation mode configured in the selected subscriber account, SAP CC generates the following tax data:

- When the taxation mode is set to “Service Provider Subject to Pay”, the tax is adjusted and added or removed to the base amount in order to determine the price amount (excluding tax) and the total price amount (including tax):
 - For postpaid accounts, tax is adjusted but not applied. SAP CC provides tax data for information to an external system that must manage or consolidate the tax data
 - For prepaid accounts, tax is both adjusted and applied

Generated Tax Data	Postpaid charged transaction		Prepaid charged transaction	
	Net	Gross	Net	Gross
Amount (Excluding Tax)	Base Amount	Base Amount - Tax Amount	Base Amount	Base Amount - Tax Amount
Tax Amount	Base Amount * Tax Rate	Base Amount * Tax Rate / (1 + Tax Rate)	Base Amount * Tax Rate	Base Amount * Tax Rate / (1 + Tax Rate)
Total Amount (Including Tax)	Base Amount + Tax Amount	Base Amount	Base Amount + Tax Amount	Base Amount
Tax Code	/<COUNTRY_CODE>/<VAT_RATE_LEVEL>#<TIMESTAMP>			
Tax Status	FOR_INFO		APPLIED	

- When the taxation mode is set to “Account Holder Subject to Pay”, the holder of the postpaid account must pay the tax to the tax authority of his residence. The tax is not applied.
 - For postpaid accounts, tax is adjusted but not applied, as the account holder must not pay the tax to the service provider but directly to his tax authority
 - For prepaid accounts, tax is not available. Do not use the value computed by the system. It is subject to change

Generated Tax Data	Postpaid charged transaction		Prepaid charged transaction	
	Net	Gross	Net	Gross
Amount (Excluding Tax)	Base Amount	Not available for gross price plan <i>Do not use the value computed by the system. It is subject to change</i>	Not available in prepaid environment. <i>Do not use the value computed by the system. It is subject to change</i>	Not available in prepaid environment. <i>Do not use the value computed by the system. It is subject to change</i>

Generated Tax Data	Postpaid charged transaction		Prepaid charged transaction	
	Net	Gross	Net	Gross
Tax Amount	Base Amount * Tax Rate <i>The account holder must pay this tax amount to the tax authority of his residence</i>	Not available for gross price plan <i>Do not use the value computed by the system. It is subject to change</i>	Not available <i>Do not use the value computed by the system. It is subject to change</i>	Not available <i>Do not use the value computed by the system. It is subject to change</i>
Total Amount (Including Tax)	Base Amount <i>The account holder must not pay the tax to the service provider</i>	Not available for gross price plan <i>Do not use the value computed by the system. It is subject to change</i>	Not available <i>Do not use the value computed by the system. It is subject to change</i>	Not available <i>Do not use the value computed by the system. It is subject to change</i>
Tax Code	/<COUNTRY_CODE>/<VAT_RATE_LEVEL>#<TIMESTAMP>			
Tax Status (*)	BUYER_SUBJECT_TO_PAY		NO_TAX	

(*) In case of error due to unsupported configuration in the master data, the tax status is equal to NO_TAX.

- When the taxation mode is set to “Account Holder Exempted”, the holder of the external (postpaid) account must not pay the tax. The tax is not applied.
 - For postpaid accounts, tax is adjusted but not applied, as the account holder must not pay the tax.
 - For prepaid accounts, tax is not available. Do not use the value computed by the system. It is subject to change

Generated Tax Data	Postpaid charged transaction		Prepaid charged transaction	
	Net	Gross	Net	Gross
Amount (Excluding Tax)	Base Amount	Not available for gross price plan <i>Do not use the value computed by the system. It is subject to change</i>	Not available <i>Do not use the value computed by the system. It is subject to change</i>	Not available <i>Do not use the value computed by the system. It is subject to change</i>
Tax Amount	Base Amount * Tax Rate <i>The account holder must not pay this tax amount to the tax authority of his residence</i>	Not available for gross price plan <i>Do not use the value computed by the system. It is subject to change</i>	Not available <i>Do not use the value computed by the system. It is subject to change</i>	Not available <i>Do not use the value computed by the system. It is subject to change</i>
Total Amount (Including Tax)	Base Amount <i>The account holder must not pay the tax to the service provider</i>	Not available for gross price plan <i>Do not use the value computed by the system. It is subject to change</i>	Not available <i>Do not use the value computed by the system. It is subject to change</i>	Not available <i>Do not use the value computed by the system. It is subject to change</i>
Tax Code	/<COUNTRY_CODE>/<VAT_RATE_LEVEL>#<TIMESTAMP>			
Tax Status (*)	TAX_EXEMPTED		NO_TAX	

(*) In case of error due to unsupported configuration in the master data, the tax status is equal to NO_TAX.

At the end of the calculation function, the computed taxes are applied on the charged transactions. These taxed & charged transactions are then returned to the calling Rater which can continue its own charging operations.

EU VAT without VAT Rules

SAP CC uses the EU VAT system depending on the tax settings in the subscriber account. When these settings are not compliant with this tax system and with the settings configured in the customized charge, SAP CC does not calculate the tax. Refer to the no tax function in the Process Functions section.

Inputs

To compute and adjust value added taxes (VAT) without VAT rules, SAP CC needs the following information:

- Information coming from the analyzed event, containing the consumption date
- Information coming from the rating operation (dynamic pricing), containing the price plan amount previously rated by SAP CC
- Information configured in the customized charge, which contains the following data:
 - A flag which defines if the base amount generated by executing the price plan are assumed to be net amounts or gross amounts
 - A tax rate level (normal, reduced, super reduced, zero tax, and so on)
- Tax information configured in the subscriber account, which contains the following data:
 - A tax code, which corresponds to the ISO 3166 country code of the tax authority (FR, DE, GB, and so on)
 - A taxation mode, which determines the entity that must pay the tax amount and thus impact the amounts according to the following rules:

Taxation mode	Impact
Service Provider Subject to Pay	Tax is calculated and added to the net amount
Account Holder Subject to Pay	Tax is calculated but not applied. The account holder must pay the tax to the tax authority of his residence.
Account Holder Exempted	Tax is calculated but not applied
None	Tax is not calculated

Tax Computing

To compute the tax data, SAP CC must compute the value of the tax rate:

- The SAP CC system retrieves the country tax policy for the selected country of the tax authority.
- The SAP CC system determines the value of the tax rate according to the tax level, the selected country tax policy, and the consumption date.

Then, the Rater computes the tax data included in the rated transactions:

Computed Tax Data in a Rated Transaction	Net Prices	Gross Prices
Tax Rate	SAP CC computes the tax rate value according to: <ul style="list-style-type: none"> ■ The tax rate level and the consumption date ■ The country tax policy configured in the subscriber account 	

Computed Tax Data in a Rated Transaction	Net Prices	Gross Prices
Rated Tax Amount	Calculates the tax amount based on the price amount computed during the execution of the price plan, the tax rate: Tax Rate * Price Plan Amount	Tax Rate * Price Plan Amount / (1 + Tax Rate)
Rated Amount (Before Tax Computing)	Price Plan Amount	Price Plan Amount
Total Rated Amount (Including Tax)	Rated Amount + Rated Tax Amount	Price Plan Amount

Tax Adjusting and Applying

This function starts when SAP CC has split a rated transaction into one or several charged transactions during the execution of the charging plan. The rated amount in the rated transaction is also split into these charged transactions as a base amount. The sum of the base amounts is equal to rated amount. Each charged transaction is assigned to an account (prepaid or external) in the subscriber account previously identified.

The tax rate of the rated transaction is computed.

The computed taxes are then adjusted. SAP CC determines who is subject to pay the tax and the corresponding tax policy. SAP CC determines the tax amount in the charged transaction depending on the base amount and the tax rate value:

Adjusted Tax Data in Charged Transactions	Net Prices	Gross Prices
Tax Amount	Base Amount * Tax Rate	Base Amount * Tax Rate / (1 + Tax Rate)

Finally SAP CC applies (or not) the tax on these charged transactions, which thus become taxed & charged transactions, ready for persistency purposes. It uses the taxation mode configured in the subscriber account to decide how to apply the tax. Depending on the taxation mode configured in the selected subscriber account, SAP CC generates the following tax data:

- When the taxation mode is set to “Service Provider Subject to Pay”, the tax is adjusted and added or removed to the base amount in order to determine the price amount (excluding tax) and the total price amount (including tax):
 - For postpaid accounts, tax is adjusted but not applied. SAP CC provides tax data for information to an external system that must manage or consolidate the tax data
 - For prepaid accounts, tax is both adjusted and applied

Generated Tax Data	Postpaid charged transaction		Prepaid charged transaction	
	Net	Gross	Net	Gross
Amount (Excluding Tax)	Base Amount	Base Amount - Tax Amount	Base Amount	Base Amount - Tax Amount

Generated Tax Data	Postpaid charged transaction		Prepaid charged transaction	
	Net	Gross	Net	Gross
Tax Amount	Base Amount * Tax Rate	Base Amount * Tax Rate / (1 + Tax Rate)	Base Amount * Tax Rate	Base Amount * Tax Rate / (1 + Tax Rate)
Total Amount (Including Tax)	Base Amount + Tax Amount	Base Amount	Base Amount + Tax Amount	Base Amount
Tax Code	/<COUNTRY_CODE>/<VAT_RATE_LEVEL>#<TIMESTAMP>			
Tax Status	FOR_INFO		APPLIED	

- When the taxation mode is set to “Account Holder Subject to Pay”, the holder of the postpaid account must pay the tax to the tax authority of his residence. The tax is not applied.
 - For postpaid accounts, tax is adjusted but not applied, as the account holder must not pay the tax to the service provider but directly to his tax authority
 - For prepaid accounts, tax is not available. Do not use the value computed by the system. It is subject to change

Generated Tax Data	Postpaid charged transaction		Prepaid charged transaction	
	Net	Gross	Net	Gross
Amount (Excluding Tax)	Base Amount	Not available for gross price plan <i>Do not use the value computed by the system. It is subject to change</i>	Not available <i>Do not use the value computed by the system. It is subject to change</i>	Not available <i>Do not use the value computed by the system. It is subject to change</i>
Tax Amount	Base Amount * Tax Rate <i>The account holder must pay this tax amount to the tax authority of his residence</i>	Not available for gross price plan <i>Do not use the value computed by the system. It is subject to change</i>	Not available <i>Do not use the value computed by the system. It is subject to change</i>	Not available <i>Do not use the value computed by the system. It is subject to change</i>
Total Amount (Including Tax)	Base Amount <i>The account holder must not pay the tax to the service provider</i>	Not available for gross price plan <i>Do not use the value computed by the system. It is subject to change</i>	Not available <i>Do not use the value computed by the system. It is subject to change</i>	Not available <i>Do not use the value computed by the system. It is subject to change</i>
Tax Code	/<COUNTRY_CODE>/<VAT_RATE_LEVEL>#<TIMESTAMP>			
Tax Status (*)	BUYER_SUBJECT_TO_PAY		NO_TAX	

(*) In case of error due to unsupported configuration in the master data, the tax status is equal to NO_TAX.

- When the taxation mode is set to “Account Holder Exempted”, the holder of the external (postpaid) account must not pay the tax. The tax is not applied.
 - For postpaid accounts, tax is adjusted but not applied, as the account holder must not pay the tax
 - For prepaid accounts, tax is not available. Do not use the value computed by the system. It is subject to change

Generated Tax Data	Postpaid charged transaction		Prepaid charged transaction	
	Net	Gross	Net	Gross
Amount (Excluding Tax)	Base Amount	Not available for gross price plan <i>Do not use the value computed by the system. It is subject to change.</i>	Not available <i>Do not use the value computed by the system. It is subject to change.</i>	Not available <i>Do not use the value computed by the system. It is subject to change.</i>
Tax Amount	Base Amount * Tax Rate <i>The account holder must not pay this tax amount to the tax authority of his residence</i>	Not available for gross price plan <i>Do not use the value computed by the system. It is subject to change.</i>	Not available <i>Do not use the value computed by the system. It is subject to change.</i>	Not available <i>Do not use the value computed by the system. It is subject to change.</i>
Total Amount (Including Tax)	Base Amount <i>The account holder must not pay the tax to the service provider.</i>	Not available for gross price plan <i>Do not use the value computed by the system. It is subject to change.</i>	Not available <i>Do not use the value computed by the system. It is subject to change.</i>	Not available <i>Do not use the value computed by the system. It is subject to change.</i>
Tax Code	/<COUNTRY_CODE>/<VAT_RATE_LEVEL>#<TIMESTAMP>			
Tax Status (*)	TAX_EXEMPTED		NO_TAX	

(*) In case of error due to unsupported configuration in the master data, the tax status is equal to NO_TAX.

At the end of the calculation function, the computed taxes are applied on the charged transactions. These taxed & charged transactions are then returned to the calling Rater which can continue its own rating operations.

Flat Tax

Inputs

To compute and adjust flat taxes, SAP CC needs the following information:

- Information coming from the rating operation (dynamic pricing), containing the price plan amount previously rated by SAP CC
- Information configured in the customized charge, which contains the following data:
 - A flag, used to define if the base amounts generated during the execution of the price plan are assumed to be net amounts or gross amounts
 - A value of flat tax rate (expressed as a percentage)
- Tax information, which is configured in the subscriber account and contains the following data:
 - A tax code, which corresponds to the ISO 3166 country code of the tax authority (FR, DE, GB, and so on)
 - A taxation mode, which determines the entity that must pay the tax amount and thus impact the amounts

Tax Computing

The Rater computes the following tax data included in the rated transactions:

Computed Tax Data in a Rated Transaction	Net Prices	Gross Prices
Rated Tax Amount	Calculates the tax amount based on the price amount computed during the execution of the price plan and on the tax rate: Flat Tax Rate * Price Plan Amount	Flat Tax Rate * Price Plan Amount / (1 + Flat Tax Rate)
Rated Amount (Before Tax Computing)	Price Plan Amount	Price Plan Amount
Total Rated Amount (Including Tax)	Rated Amount + Rated Tax Amount	Price Plan Amount

Tax Adjusting and Applying

This function starts when the SAP CC system has split a rated transaction to one or several charged transactions during the execution of the charging plan. The rated amount in the rated transaction is also split to these charged transactions as a base amount. The sum of the base amounts is equal to rated amount. Each charged transaction is assigned to an account (prepaid or external) in the subscriber account previously identified.

SAP CC uses the flat tax rate of the rated transaction is used to adjust the computed taxes are then adjusted. The SAP CC system determines who is subject to pay the tax and the corresponding tax policy.

The SAP CC system determines the tax amount in the charged transaction depending on the base amount and the value of the flat tax rate:

Adjusted Tax Data in Charged Transactions	Net Prices	Gross Prices
Tax Amount	Base Amount * Flat Tax Rate	Base Amount * Flat Tax Rate / (1 + Flat Tax Rate)

Finally, SAP CC applies (or not) the tax on these charged transactions, which thus become taxed & charged transactions, ready for persistency purposes. It uses the taxation mode configured in the subscriber account to decide how to apply the tax. Depending on the taxation mode configured in the selected subscriber account, SAP Convergent Charging generates the following tax data:

- When the taxation mode is set to “Service Provider Subject to Pay”, the tax is adjusted and added or removed to the base amount in order to determine the price amount (excluding tax) and the total price amount (including tax).
 - For postpaid accounts, tax is both adjusted and applied. SAP CC provides tax data for information to an external system that must manage or consolidate the tax data
 - For prepaid accounts, tax is both adjusted and applied

Generated Tax Data	Postpaid charged transaction		Prepaid charged transaction	
	Net	Gross	Net	Gross
Amount (Excluding Tax)	Base Amount	Base Amount - Tax Amount	Base Amount	Base Amount - Tax Amount
Tax Amount	Base Amount * Flat Tax Rate	Base Amount * Flat Tax Rate / (1 + Flat Tax Rate)	Base Amount * Flat Tax Rate	Base Amount * Flat Tax Rate / (1 + Flat Tax Rate)

Generated Tax Data	Postpaid charged transaction		Prepaid charged transaction	
	Net	Gross	Net	Gross
Total Amount (Including Tax)	Base Amount + Tax Amount	Base Amount	Base Amount + Tax Amount	Base Amount
Tax Code	/RAW/<LABEL_IN_CUSTOMIZED_CHARGE>			
Tax Status	APPLIED			

- When the taxation mode is set to “Account Holder Subject to Pay”, the holder of the postpaid account must pay the tax to the tax authority of his residence. The tax is not applied.
 - For postpaid accounts, tax is adjusted but not applied, as the account holder must not pay the tax to the service provider but directly to his tax authority
 - For prepaid accounts, tax is not available. Do not use the value computed by the system. It is subject to change

Generated Tax Data	Postpaid charged transaction		Prepaid charged transaction	
	Net	Gross	Net	Gross
Amount (Excluding Tax)	Base Amount	Not available for gross price plan <i>Do not use the value computed by the system. It is subject to change.</i>	Not available <i>Do not use the value computed by the system. It is subject to change.</i>	Not available <i>Do not use the value computed by the system. It is subject to change.</i>
Tax Amount	Base Amount * Flat Tax Rate <i>The account holder must pay this tax amount to the tax authority of his residence</i>	Not available for gross price plan <i>Do not use the value computed by the system. It is subject to change.</i>	Not available <i>Do not use the value computed by the system. It is subject to change.</i>	Not available <i>Do not use the value computed by the system. It is subject to change.</i>
Total Amount (Including Tax)	Base Amount <i>The account holder must not pay the tax to the service provider.</i>	Not available for gross price plan <i>Do not use the value computed by the system. It is subject to change.</i>	Not available <i>Do not use the value computed by the system. It is subject to change.</i>	Not available <i>Do not use the value computed by the system. It is subject to change.</i>
Tax Code	/RAW/<LABEL_IN_CUSTOMIZED_CHARGE>			
Tax Status (*)	BUYER_SUBJECT_TO_PAY		NO_TAX	

(*) In case of error due to unsupported configuration in the master data, the tax status is equal to NO_TAX.

- When the taxation mode is set to “Account Holder Exempted”, the holder of the external (postpaid) account must not pay the tax. The tax is not applied.
 - For postpaid accounts, tax is adjusted but not applied, as the account holder must not pay the tax
 - For prepaid accounts, tax is not available. Do not use the value computed by the system. It is subject to change

Generated Tax Data	Postpaid charged transaction		Prepaid charged transaction	
	Net	Gross	Net	Gross
Amount (Excluding Tax)	Base Amount	Not available for gross price plan <i>Do not use the value computed by the system. It is subject to change.</i>	Not available <i>Do not use the value computed by the system. It is subject to change.</i>	Not available <i>Do not use the value computed by the system. It is subject to change.</i>
Tax Amount	Base Amount * Tax Rate <i>The account holder must not pay this tax amount to the tax authority of his residence</i>	Not available for gross price plan <i>Do not use the value computed by the system. It is subject to change.</i>	Not available <i>Do not use the value computed by the system. It is subject to change.</i>	Not available <i>Do not use the value computed by the system. It is subject to change.</i>
Total Amount (Including Tax)	Base Amount <i>The account holder must not pay the tax to the service provider.</i>	Not available for gross price plan <i>Do not use the value computed by the system. It is subject to change.</i>	Not available <i>Do not use the value computed by the system. It is subject to change.</i>	Not available <i>Do not use the value computed by the system. It is subject to change.</i>
Tax Code	/RAW/<LABEL_IN_CUSTOMIZED_CHARGE>			
Tax Status (*)	TAX_EXEMPTED		NO_TAX	

(*) In case of error due to unsupported configuration in the master data, the tax status is equal to NO_TAX.

At the end of the calculation function, the computed taxes are applied on the charged transactions. These taxed & charged transactions are then returned to the calling Rater which can continue its own rating operations.

U.S. Telco Tax (EZTax Tax System)

Inputs

By default, SAP CC uses the BillSoft EZTax tax system to calculate taxes related to the U.S. Telco taxes. This tax system is installed in the system landscape. Taxer instances can delegate tax calculation operations to this tax system.

To compute and adjust U.S. Telco taxes, SAP CC needs the following information:

- Information coming from the analyzed event, containing the consumption date
- Information coming from the rating operation (dynamic pricing), containing the price plan amount previously rated by SAP CC
- Tax information configured in the customized charge and containing the following data:

Information	EZTax Property
Taxing jurisdiction information	The correct jurisdiction to the transaction for obtaining and returning the correct tax to the billing system: <ul style="list-style-type: none"> ■ Origination ■ Termination ■ Service address

Information	EZTax Property
Customer information (subscriber)	<ul style="list-style-type: none"> ■ Customer type ■ Incorporated city area ■ Exemption ■ Service class and business class ■ Lifeline
Company information (service provider)	<ul style="list-style-type: none"> ■ Reseller ■ Facilities based ■ Franchise
Transaction information	<ul style="list-style-type: none"> ■ Number of lines ■ Number of locations ■ Client resale ■ Telecommunication type ■ Call duration ■ Transaction service type

- Tax information configured in the subscriber account and containing the following data:

Information	EZTax Property
Taxing jurisdiction information	<p>The correct jurisdiction to the transaction for obtaining and returning the correct tax to the billing system:</p> <ul style="list-style-type: none"> ■ Origination ■ Termination ■ Service address
Customer information (subscriber)	<ul style="list-style-type: none"> ■ Customer type ■ Incorporated city area
Transaction information	<ul style="list-style-type: none"> ■ Client resale



NOTES

- For further information about the BillSoft EZTax tax system, refer to its product documentation
- For further information about these EZTax properties, refer to the [SAP Convergent Charging Core Tool Online Help](#) documentation
- The tax journal can be updated:
 - Before returning taxed & charged transactions to the Rater, which might lead to gaps with reality if the third-party billing system decides to exclude some elements of the invoicing process
 - After returning the taxed & charged transactions to the Rater, when a confirmation is received from the third-party billing system. This delayed update ensures that all the elements included in the invoicing process are mentioned in the tax journal

Tax Computing

The Rater computes the following tax data included in the rated transactions:

Computed Tax Data in a Rated Transaction	Net Prices
Rated Tax Amount	Global Tax Amount The tax amount is retrieved from the BillSoft EZTax tax system.
Rated Amount (Before Tax Computing)	Price Plan Amount The rated amount is the price amount determined during the execution of the price plan.
Total Rated Amount (Including Tax)	Rated Amount + Rated Tax Amount The total rated amount is the sum of the tax amount and the price amount determined during the execution of the price plan.

Tax Adjusting and Applying

SAP CC adjusts the computed tax amount of the original rated transaction to a prorated tax amount depending on the “global amount/charged amount” ratio.



NOTE

The tax amount is not re-calculated for each charged transaction. Each amount of tax associated to a charged transaction is prorated depending on the “global amount/charged amount” ratio.

SAP CC applies the computed taxes on the charged transactions. SAP CC determines the tax amount in the charged transaction depending on the base amount and the tax rate value:

Adjusted Tax Data in Charged Transactions	Net Prices
Tax Amount	$\text{Base Amount} * \text{Rated Tax Amount} / \text{Total Rated Amount}$

These taxed & charged transactions are then returned to the calling Rater which can continue its own rating operations.



NOTE

The overrun mechanism can produce as many charged transactions as concerned/impacted accounts which must be charged, but the tax amount is calculated on the global transaction. This guarantees that the tax authority collects the correct tax amount, no matter if this amount is split or not.

Depending on the taxation mode configured in the selected subscriber account, SAP CC generates the following tax data:

- When the taxation mode is set to “Service Provider Subject to Pay”, the tax is adjusted and added or removed to the base amount in order to determine the price amount (excluding tax) and the total price amount (including tax):
 - For postpaid accounts, tax is both adjusted and applied. SAP CC provides tax data for information to an external system which must manage or consolidate the tax data
 - For prepaid accounts, tax is both adjusted and applied.

Generated Tax Data	Postpaid charged transaction		Prepaid charged transaction	
	Net	Gross	Net	Gross
Amount (Excluding Tax)	Base Amount	Not available <i>Do not use the value computed by the system. It is subject to change.</i>	Base Amount	Not available <i>Do not use the value computed by the system. It is subject to change.</i>
Tax Amount	Base Amount * Rated Tax Amount / Total Rated Amount	Not available <i>Do not use the value computed by the system. It is subject to change.</i>	Base Amount * Rated Tax Amount / Total Rated Amount	Not available <i>Do not use the value computed by the system. It is subject to change.</i>
Total Amount (Including Tax)	Base Amount + Tax Amount	Not available <i>Do not use the value computed by the system. It is subject to change.</i>	Base Amount + Tax Amount	Not available <i>Do not use the value computed by the system. It is subject to change.</i>
Tax Code	EZTAXED			
Tax Status	APPLIED	N/A	APPLIED	N/A

- When the taxation mode is set to “Account Holder Subject to Pay”, the holder of the postpaid account must pay the tax to the tax authority of his residence. The tax is not applied.
 - For postpaid accounts, tax is adjusted but not applied, as the account holder must not pay the tax to the service provider but directly to his tax authority
 - For prepaid accounts, tax is not available. Do not use the value computed by the system. It is subject to change.

Generated Tax Data	Postpaid charged transaction		Prepaid charged transaction	
	Net	Gross	Net	Gross
Amount (Excluding Tax)	Base Amount	Not available <i>Do not use the value computed by the system. It is subject to change.</i>	Not available <i>Do not use the value computed by the system. It is subject to change.</i>	Not available <i>Do not use the value computed by the system. It is subject to change.</i>
Tax Amount	Base Amount * Rated Tax Amount / Total Rated Amount <i>The account holder must pay this tax amount to the tax authority of his residence</i>	Not available <i>Do not use the value computed by the system. It is subject to change.</i>	Not available <i>Do not use the value computed by the system. It is subject to change.</i>	Not available <i>Do not use the value computed by the system. It is subject to change.</i>
Total Amount (Including Tax)	Base Amount <i>The account holder must not pay the tax to the service provider.</i>	Not available <i>Do not use the value computed by the system. It is subject to change.</i>	Not available <i>Do not use the value computed by the system. It is subject to change.</i>	Not available <i>Do not use the value computed by the system. It is subject to change.</i>
Tax Code	EZTAXED			
Tax Status (*)	BUYER_SUBJECT_TO_PAY		Not available or no tax	

(*) In case of error due to unsupported configuration in the master data, the tax status is equal to NO_TAX.

- When the taxation mode is set to “Account Holder Exempted”, the holder of the external (postpaid) account must not pay the tax. The tax is not applied.
 - For postpaid accounts, tax is adjusted but not applied, as the account holder must not pay the tax
 - For prepaid accounts, tax is not available. Do not use the value computed by the system. It is subject to change

Generated Tax Data	Postpaid charged transaction		Prepaid charged transaction	
	Net	Gross	Net	Gross
Amount (Excluding Tax)	Base Amount	Not available <i>Do not use the value computed by the system. It is subject to change.</i>	Not available <i>Do not use the value computed by the system. It is subject to change.</i>	Not available <i>Do not use the value computed by the system. It is subject to change.</i>
Tax Amount	Base Amount * Rated Tax Amount / Total Rated Amount <i>The account holder must pay this tax amount to the tax authority of his residence</i>	Not available <i>Do not use the value computed by the system. It is subject to change.</i>	Not available <i>Do not use the value computed by the system. It is subject to change.</i>	Not available <i>Do not use the value computed by the system. It is subject to change.</i>
Total Amount (Including Tax)	Base Amount <i>The account holder must not pay the tax to the service provider.</i>	Not available <i>Do not use the value computed by the system. It is subject to change.</i>	Not available <i>Do not use the value computed by the system. It is subject to change.</i>	Not available <i>Do not use the value computed by the system. It is subject to change.</i>
Tax Code	EZTAXED			
Tax Status (*)	TAX_EXEMPTED		Not available or no tax	

(*) In case of error due to unsupported configuration in the master data, the tax status is equal to NO_TAX.

These taxed & charged transactions are then returned to the calling Rater which can continue its own rating operations.

No tax system

If tax is not enabled neither in the subscriber account nor in the customized charge, SAP CC does not compute any tax data:

- Tax is disabled in customized charges
- EU VAT system is configured in the customized charge but the tax rate level is set to no tax
- EU VAT system is selected in the subscriber account but the taxation mode is set to none
- EU VAT system is selected in the subscriber account with a taxation mode but the tax zone is set to none

NOTE

Degraded mode: When the tax calculation is not possible, SAP CC does not compute tax data. The tax settings in the charge plan are not compliant with the settings in the subscriber account.

Tax Computing

Tax is not computed. SAP CC generates the following data:

Computed Tax Data in a Rated Transaction	Net Prices	Gross Prices
Rated Tax Amount	0	0
Rated Amount (Before Tax Computing)	Price Plan Amount	Price Plan Amount
Total Rated Amount (Including Tax)	Price Plan Amount	Price Plan Amount

Tax Adjusting and Applying

Tax is not computed. SAP CC generates the following data:

Tax Data in Charged Transactions	Postpaid charged transaction		Prepaid charged transaction	
	Net	Gross	Net	Gross
Amount (Excluding Tax)	Base Amount			
Tax Amount	0			
Total Amount (Including Tax)	Base Amount			
Tax Code	NO_TAX			
Tax Status	No tax for the transaction : NO_TAX			

Process Limitations

Overrun with U.S. Telco Taxes (only relevant for the CC 2.0 and CC 3.0 model)

- The overrun mechanism can produce as many charged transactions as concerned/impacted accounts which must be charged, but the tax amount is calculated on the global transaction. This guarantees that the tax authority collects the correct tax amount, no matter if this amount is split or not.
- The tax amount is not re-calculated for each charged transaction. Each amount of tax associated to a charged transaction is prorated depending on the “global amount/charged amount” ratio.
- The tax amounts calculated by the U.S. Telco Taxes tax system can be nonlinear (e.g. function of a quantity, a duration, and so on). As a consequence, each tax amount can be wrong in terms of tax reporting if these amounts are considered individually, but the total amount of tax is correct.

Session-based charging process with U.S. Telco Taxes

- The session-based charging process creates multiple charged transactions which are individually taxed
- The session-based charging process does not create any global transaction at the end of a charging session
- The tax amounts calculated by the U.S. Telco Taxes tax system can be nonlinear (e.g. function of a quantity, a duration, and so on). When such a situation occurs, each tax amount associated to the transactions generated during a charging session might not be correct. The sum of these tax amounts might be slightly different from the tax which would have been associated to a global transaction

Operations on external references (only relevant for the subscriptions data model)

All operations related to external references do not calculate any tax as the SAP Convergent Charging solution does not have access to the required tax information.

Commissions or sponsorship (only relevant for the CC 2.0 and CC 3.0 model)

In case of commissioning or sponsorship, it is recommended to use an external reference in the charging plan of the dependent charge instead of an internal reference to an external account. As a consequence and due to the above explanation, no tax are computed in commissioning or sponsorship situations.

Credit Control Messages Conversion Process

Process Description

The Credit Control Conversion process is used to make the conversion between credit control messages (CCRs and CCAs) and SAP CC charging messages. This conversion both concerns:

- The CC-Request-Type and Requested-Action AVPs of the credit control request, which are used to determine the SAP CC charging operation to perform
- The list of AVPs included into the credit control request or response and which correspond or must be mapped to the properties of an SAP CC charging message. These AVPs:
 - May be declared into an AVP Dictionary
 - Accessible through a given AVP Path, a CSV list of AVP codes. When this path contains codes which refer to multiple occurrences, square brackets ([]) are used to point to the adequate node.



EXAMPLE

440[0],442: This AVP path gives the possibility to access to the unique child (Service-Parameter-Value , code: 442) of the first occurrence of the Service-Parameter-Info AVP (code: 440).

AVP Dictionary

The AVP Dictionary contains the descriptions of the AVPs used in CCRs and CCAs, which can be:

- Old standard AVPs
- Vendor-specific AVPs

These AVPs are not natively recognized by the Diameter Stack provided by the Traffix Systems company, and must thus be defined and described to be used during CCRs and CCAs mapping operations.

The AVP Dictionary is based on the following XSD fragments:

```
<xs:element name="avpDictionary">
  <xs:complexType>
    <xs:attribute name="version" type="xs:string" use="required"/>
    <xs:attribute name="date" type="xs:dateTime"/>
    <xs:sequence>
      <xs:element ref="avpDesc" minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

With the following substructures:

```
<xs:element name="avpDesc">
  <xs:complexType>
    <xs:attribute name="name" type="xs:string"/>
    <xs:attribute name="code" type="xs:integer" use="required"/>
    <xs:attribute name="type" type="AvpType" use="required"/>
    <xs:attribute name="vendorId" type="xs:long" use="required"/>
  </xs:complexType>
```

```

</xs:element>
<xs:simpleType name="AvpType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="DiameterURI"/>
    <xs:enumeration value="DiameterIdentity"/>
    <xs:enumeration value="Address"/>
    <xs:enumeration value="UTF8String"/>
    <xs:enumeration value="OctetString"/>
    <xs:enumeration value="Unsigned32"/>
    <xs:enumeration value="Unsigned64"/>
    <xs:enumeration value="Time"/>
    <xs:enumeration value="Integer32"/>
    <xs:enumeration value="Integer64"/>
    <xs:enumeration value="Float32"/>
    <xs:enumeration value="Float64"/>
    <xs:enumeration value="Enumerated"/>
    <xs:enumeration value="Grouped"/>
  </xs:restriction>
</xs:simpleType>

```

**EXAMPLE**

The following code represents an AVP dictionary containing both standard AVPs (the “10415” identifier corresponds to a 3GPP AVP) and vendor-specific AVPs (the “11” identifier only concerns the Hewlett Packard company):

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- Specific AVP Dictionary for Diameter server -->
<avpDictionary version="1.0" date="2008-01-01T00:00:00">
  <avpDesc name="Service-Information" code="873" type="Grouped" vendorId="10415"/>
  <avpDesc name="Service-Generic-Information" code="1256" type="Grouped" vendorId="10415"/>
  <avpDesc name="Deny-Reason" code="10027" type="Enumerated" vendorId="11"/>
  <avpDesc name="Play-Announcement-Id" code="10045" type="Unsigned32" vendorId="11"/>
  <avpDesc name="Subscriber-Profile" code="10008" type="Grouped" vendorId="11"/>
  <avpDesc name="Subscriber-Language" code="10009" type="UTF8String" vendorId="11"/>
</avpDictionary>

```

Service Dictionary

The mapping between the AVPs and the attributes of the charging message is declared into a XML structure managed by the Diameter Server. This structure is named Service Dictionary and gives the possibility to register multiple services with different mappings. This dictionary is based on service keys which must correspond to the service identifier provided in the CCR.

**NOTE**

The AVP used to define the service key must be included in all CCRs sent to the Diameter Server. If this AVP is not provided, a CCA with the DIAMETER_UNABLE_TO_COMPLY error is returned.

The Service Dictionary is based on the following XSD fragments:

```

<xs:element name="serviceDictionary" type="cs:serviceDictionary" />
<xs:complexType name="serviceDictionary">
  <xs:sequence>
    <xs:element ref="cs:serviceKeyDesc" minOccurs="1" maxOccurs="1" />
    <xs:element ref="cs:abstractServiceDesc" minOccurs="1" maxOccurs="unbounded" />
  </xs:sequence>
  <xs:attribute name="version" type="xs:string" use="required" />
  <xs:attribute name="date" type="xs:dateTime" use="required" />
</xs:complexType>

```

With the following substructures:

```
<xs:element name="serviceKeyDesc" type="cs:serviceKeyDescription"
  substitutionGroup="cs:abstractServiceKeyDesc" />
<xs:element name="abstractServiceKeyDesc" type="cs:serviceKeyDescription" />
<xs:complexType name="serviceKeyDescription">
  <xs:sequence>
    <xs:element ref="cs:keyDesc" minOccurs="1" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
<xs:element name="keyDesc" type="cs:keyDescription" />
<xs:complexType name="keyDescription">
  <xs:attribute name="name" type="xs:string" use="required" />
  <xs:attribute name="avpPath" type="cs:AvpPath" use="required" />
</xs:complexType>
```



EXAMPLE

Considering an implementation where the Call-Type AVP is used as a service key:

```
<serviceKeyDesc>
  <keyDesc name="Call-Type" avpPath="873,901,943"/>
</serviceKeyDesc>
```

This AVP identifies the type of the service and its direction. It is nested in the specific CS-Information AVP (code: 901) part of the Service-Information AVP (code: 873), and can take the following values:

- | | |
|-----------------------------------|--------------------|
| ■ MO_voice | ■ MO_data |
| ■ MT_voice | ■ MT_data |
| ■ MT_voice_forwarded_to_number | ■ Three_way_call |
| ■ MT_voice_forwarded_to_voicemail | ■ Conference_call |
| ■ MO_fax | ■ MF_voice_regular |
| ■ MT_fax | ■ MF_fax |
| ■ MO_SMS | ■ MF_GPRS |
| ■ MT_SMS | ■ MF_data |
| ■ MO_GPRS | ■ MO_USSD |
| ■ MT_GPRS | |

Once the service has been identified, the Service Dictionary is then parsed to retrieve the associated request and response mappings. These mappings are located in the Service Description which is based on the following XSD fragments:

```
<xs:element name="serviceDesc" type="cs:requestServiceDescription"
  substitutionGroup="cs:abstractServiceDesc" />
<xs:complexType name="requestServiceDescription">
  <xs:complexContent>
    <xs:extension base="cs:serviceDescription">
      <xs:sequence>
        <xs:element ref="cs:requestMappingDesc" minOccurs="1" maxOccurs="1" />
        <xs:element ref="cs:answerMappingDesc" minOccurs="0" maxOccurs="1" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

With the following substructures:

```
<xs:element name="abstractServiceDesc" type="cs:serviceDescription" />
<xs:complexType name="serviceDescription" abstract="true">
  <xs:sequence>
```

```

    <xs:element ref="cs:serviceKey" minOccurs="1" maxOccurs="1" />
  </xs:sequence>
  <xs:attribute name="reservationTTL" type="xs:long" use="required" />
  <xs:attribute name="reservationExpirationResolution" type="cs:ReservationExpirationResolution"
    use="required" />
</xs:complexType>

<xs:element name="requestMappingDesc" type="cs:requestMappingDescription"
  substitutionGroup="cs:abstractMappingDesc" />
<xs:complexType name="requestMappingDescription">
  <xs:complexContent>
    <xs:extension base="cs:mappingDescription">
      <xs:sequence>
        <xs:element ref="cs:serviceUserIdDesc" minOccurs="1" maxOccurs="1" />
        <xs:element ref="cs:serviceIdDesc" minOccurs="1" maxOccurs="1" />
        <xs:choice>
          <xs:sequence>
            <xs:element ref="cs:eventNameDesc" minOccurs="1" maxOccurs="1" />
            <xs:element ref="cs:unitPropertyDesc" minOccurs="1" maxOccurs="1" />
            <xs:element ref="cs:infoPropertyDesc" minOccurs="0" maxOccurs="unbounded" />
          </xs:sequence>
          <xs:element ref="cs:mccServiceKeyDesc" minOccurs="1" maxOccurs="1" />
        </xs:choice>
      </xs:sequence>
    </xs:extension>
  </xs:complexType>

<xs:element name="answerMappingDesc" type="cs:answerMappingDescription" />

<xs:complexType name="answerMappingDescription">
  <xs:attribute name="className" type="xs:string" use="required" />
</xs:complexType>

```

 **NOTE**

Every substructure of the Service Description described above is made up with dedicated substructures and thus represents a more complex structure. For further information about Diameter service dictionaries, refer to the Diameter RFC specifications.

And with the following substructures in case of multiservice sessions:

```

<xs:element name="mccServiceDesc"
  type="cs:mccServiceDescription"
  substitutionGroup="cs:abstractServiceDesc" />

<xs:complexType name="mccServiceDescription">
  <xs:complexContent>
    <xs:extension base="cs:serviceDescription">
      <xs:sequence>
        <xs:element ref="cs:mccMappingDesc" minOccurs="1" maxOccurs="1" />
      </xs:sequence>
    </xs:extension>
  </xs:complexType>

<xs:element name="mccMappingDesc"
  type="cs:mccMappingDescription"
  substitutionGroup="cs:abstractMappingDesc" />

<xs:complexType name="mccMappingDescription">
  <xs:complexContent>
    <xs:extension base="cs:mappingDescription">
      <xs:sequence>
        <xs:element ref="cs:eventNameDesc" minOccurs="1" maxOccurs="1" />
        <xs:element ref="cs:unitPropertyDesc" minOccurs="1" maxOccurs="1" />
        <xs:element ref="cs:infoPropertyDesc" minOccurs="0" maxOccurs="unbounded" />
        <xs:element ref="cs:reservationIdDesc" minOccurs="1" maxOccurs="1" />
      </xs:sequence>
    </xs:extension>
  </xs:complexType>

```

```

</xs:extension>
</xs:complexContent>
</xs:complexType>

```



EXAMPLES

Example 1: Voice service with the following used AVPs:

- Call-Type (as service key)
- Subscription-Id data
- Calling-Party-Address
- Called-Party-Address
- CC-Time

```

<serviceDesc reservationTTL="300" reservationExpirationResolution="cancelled">
  <serviceKey>
    <key name="Call-Type" value="MO_voice"/>
  </serviceKey>

  <requestMappingDesc>
    <serviceUserIdDesc type="END_USER_IMSI"/>
    <serviceIdDesc defaultId="voice"/>
    <eventNameDesc defaultName="MO_voice"/>
    <unitPropertyDesc name="call_time" avpName="CC-Time" requestedQuota="60"/>
    <infoPropertyDesc name="calling_number" avpPath="873,876,831"/>
    <infoPropertyDesc name="called_number" avpPath="873,876,832"/>
  </requestMappingDesc>
</serviceDesc>

```

Example 2: SMS service with the following used AVPs:

- Call-Type (as service key)
- Subscription-Id data
- CC-Input-Octets

```

<serviceDesc reservationTTL="300" reservationExpirationResolution="cancelled">
  <serviceKey>
    <key name="Call-Type" value="MO_SMS"/>
  </serviceKey>

  <requestMappingDesc>
    <serviceUserIdDesc type="END_USER_E164"/>
    <serviceIdDesc defaultId="SMS"/>
    <eventNameDesc defaultName="MO_SMS"/>
    <unitPropertyDesc name="size" avpName="CC-Input-Octets"/>
  </requestMappingDesc>
</serviceDesc>

```

Example 3: Call service with the following used AVPs:

- Call-Type (as service key)
- Rating-Group and Call-Type (as MSCC service key)
- CC-Time
- ReservationId: AVP 432 for each MSCC

```

<serviceDesc reservationTTL="300" reservationExpirationResolution="cancelled">
  <serviceKey>

```

```

    <key name="Call-Type" value="CALL" />
  </serviceKey>
  <requestMappingDesc>
    <serviceUserIdDesc type="END_USER_IMSI" />
    <serviceIdDesc defaultId="CALL" />
    <msccServiceKeyDesc>
      <keyDesc name="Rating-Group" avpPath="456[*],432" />
      <keyDesc name="Call-Type" avpPath="942,943" />
    </msccServiceKeyDesc>
  </requestMappingDesc>
</serviceDesc>

<msccServiceDesc reservationTTL="300" reservationExpirationResolution="cancelled">
  <serviceKey>
    <key name="Rating-Group" value="0" />
    <key name="Call-Type" value="CALL" />
  </serviceKey>
  <msccMappingDesc>
    <eventNameDesc defaultName="Voice" />
    <unitPropertyDesc name="duration" avpName="CC-Time" requestedQuota="60" />
    <reservationIdDesc avpPath="456[*],432" />
  </msccMappingDesc>
</msccServiceDesc>

```

CCRs Mapping

Each incoming CCR must be analyzed in order to build a charging message which can be handled by the Core Server.

The first step of this operation consists in analyzing the incoming CC-Request-Type and Requested-Action AVPs of the CCR in order to determine the corresponding charging operation. The following table shows the different charging operations which can be performed:

CC-Request-Type	Requested-Action	Charging operation
EVENT_REQUEST	DIRECT_DEBITING	Chargeable Items Charging
EVENT_REQUEST	CHECK_BALANCE	Chargeable Items Charging (blank mode)
EVENT_REQUEST	PRICE_ENQUIRY	Events Pricing of Chargeable Items Charging (blank mode)
INITIAL_REQUEST	N/A	Start of Chargeable Items Charging (session-based mode)
UPDATE_REQUEST	N/A	Update of Chargeable Items Charging (session-based mode)
TERMINATION_REQUEST	N/A	Stop of Chargeable Items Charging (session-based mode)

Once the charging operation has been determined, some other AVPs of the CCR are analyzed in order to build the adequate properties of the charging message. These AVPs can be:

- Diameter standard AVPs (coming from RFC 4006 and 3GPP)
- Vendor specific AVPs (also named VSA)

CCAs Mapping

According to the result of the charging operation, two different types of answers can be generated by the Diameter Server to the client application who previously requested credit authorization:

- Successful CCAs
- Erroneous CCAs



NOTE

These CCAs can contain both Diameter standard AVPs and Vendor-specific AVPs.

Successful CCAs

Successful CCAs contain AVPs such as:

- **Result-Code**, set to DIAMETER_SUCCESS
- **CC-Request-Type**, which corresponds to the value provided in the CCR
- **CC-Request-Number**, which corresponds to the value provided in the CCR
- **Granted-Service-Unit** in **Multiple-Services-Credit-Control**, which corresponds to the value provided in the CCR
- **Cost-Information**, which is the amount computed by the Core Server
- **Validity-Time**, which corresponds to the reservation time to live
- **CC-Session-Failover**, which corresponds to the failover mode retrieved from the configuration of the Diameter Server

Erroneous CCAs

When an error occurs during a charging operation, the Result-Code AVP of the returned CCA is set to a code which gives the possibility to identify the error. The following table shows the possible error codes, with their related descriptions:

Charging exception	Description and returned error codes
CommunicationFailureException	A network issue occurred. <ul style="list-style-type: none"> ■ DIAMETER_UNABLE_TO_COMPLY
IllegalArgumentException	An argument of the charging operation is invalid. <ul style="list-style-type: none"> ■ DIAMETER_UNABLE_TO_COMPLY
ServerFailureException	The Core Server cannot handle the charging message. <ul style="list-style-type: none"> ■ DIAMETER_TOO_BUSY: The message queue is full or a timeout occurred on the server side ■ DIAMETER_UNABLE_TO_COMPLY: Any other server reason
InvalidItemException	The usage event cannot be rated. <ul style="list-style-type: none"> ■ DIAMETER_RATING_FAILED
ForbiddenChargeException	The charging operation is not allowed. <ul style="list-style-type: none"> ■ DIAMETER_CREDIT_LIMIT_REACHED: The account balance cannot be charged ■ DIAMETER_USER_UNKNOWN: The related access or subscription cannot be found ■ DIAMETER_END_USER_SERVICE_DENIED: Authorizations are missing or related account is closed or locked or blocked ■ DIAMETER_UNABLE_TO_COMPLY: Any other reason
TransactionClearingException	The Core Server cannot clear at least one of the generated transactions. <ul style="list-style-type: none"> ■ DIAMETER_UNABLE_TO_COMPLY
StartRateException	The Core Server cannot create the required charging session. <ul style="list-style-type: none"> ■ DIAMETER_OUT_OF_SPACE: Not enough available space ■ DIAMETER_UNABLE_TO_COMPLY: Any other reason
UpdateRateException	The Core Server cannot update the related charging session. <ul style="list-style-type: none"> ■ DIAMETER_UNKNOWN_SESSION_ID: Unable to find the session

Charging exception	Description and returned error codes
	<ul style="list-style-type: none"> ■ DIAMETER_OUT_OF_SPACE: Not enough available space ■ DIAMETER_UNABLE_TO_COMPLY: Any other reason
StopRateException	<p>The Core Server cannot stop the related charging session.</p> <ul style="list-style-type: none"> ■ DIAMETER_UNKNOWN_SESSION_ID: Unable to find the session ■ DIAMETER_OUT_OF_SPACE: Not enough available space ■ DIAMETER_UNABLE_TO_COMPLY: Any other reason

 **NOTE**

When charging operations are managed through sessions, every error automatically implies the deletion of the associated session.

Custom AVPs in CCAs

As described above, it is possible to include additional AVPs (vendor-specific or not) in the CCAs. To map a property to an AVP of the CCA, it is necessary to:

- Update the “answerMappingDesc” section of the AVP dictionary in order to specify a the name of a Java class used to add vendor-specific attributes to the CCA
- Create this Java class, which must implement the “AnswerMapping” provided Java interface

 **EXAMPLE**

The following code illustrates the management of a vendor-specific AVP named “Deny-Reason”. This VSA is used to provide the reason why a service is denied to a given subscriber (which is associated to the DIAMETER_END_USER_SERVICE_DENIED error). This VSA can be set to the following values:

- SUBSCRIBER_INACTIVE (0)
- ORIGIN_PROHIBITED (1)
- DESTINATION_PROHIBITED (2)
- SUBSCRIBER_NOT_PREPAID (3)

When the service is denied, SAP CC returns a java exception named “ForbiddenChargeException” containing a reason which must be mapped to an attribute of the CCA. This mapping is performed in the “onEndUserServiceDenied” method provided by the “AnswerMapping” Java interface:

```
public void onEndUserServiceDeniedError(RoAnswerContext roAnswerContext,
                                       OperationFailureException chargingException) {

    // handle Deny-Reason AVP
    ForbiddenChargeException fce = (ForbiddenChargeException) chargingException;
    int denyReason = 2;
    switch (fce.getError()) {
        case ForbiddenChargeException.NOT_AUTHORIZED: // no access component
            String message = fce.getMessage();
            if (message.indexOf(ORIGIN_PROHIBITED) > -1) {
                denyReason = 1;
            } else if (message.indexOf(DESTINATION_PROHIBITED) > -1) {
                denyReason = 2;
            }
            break;
        case ForbiddenChargeException.ACCOUNT_BLOCKED:
        case ForbiddenChargeException.ACCOUNT_LOCKED:
        case ForbiddenChargeException.ACCOUNT_CLOSED:
            denyReason = 0;
    }
}
```

```
        break;
    }

    // add Service-Information (873)
    GroupedVSAvp gavp = VSAvpFactory.getInstance().createGroupedVSAvp(873);
    AvpList currentList = roAnswerContext.getCommandLevelAvpList().addGroupedVSAvp(gavp);

    // add Service-Generic-Information (1256)
    gavp = VSAvpFactory.getInstance().createGroupedVSAvp(1256);
    AvpList sgi = currentList.addGroupedVSAvp(gavp);

    // add Deny-Reason (10027) enumerated
    VSAvp avp = VSAvpFactory.getInstance().createVSAvp(10027, new BigDecimal(denyReason));
    sgi.addVSAvp(avp);
}
```

Process Limitations

SAP CC Diameter Server component does not support:

- The Re-Authorization Request (RAR) messages. It cannot handle the renewal reservation notifications managed by the SAP CC Core Server component
- The Sy command messages: Spending-Limit-Request (SLR) and Spending-Status-Notification-Request (SNR). It cannot handle the spending status reports managed by the SAP CC Core Server component

8 Procedures

Deploying SAP CC

Securing SAP CC

As described in the Security Guide documentation, the communication between the deployed technical components of SAP CC relies on the following protocols:

- HTTP
 - Used to transport SOAP messages (*Web Services* technical interface)
 - Used to transport proprietary XML messages (Http Communication Interface)
 - Used by the SMD system when executing the CCDBProviderWS web service
 - Used to send payload data to the SLD system (which might be itself secured)
- TCP/IP
 - Used to transport proprietary TCP packets
 - Used to transport RFC calls when communicating with SAP CI
- UDP, used to transport messages dedicated to network discovery purposes
- JDBC, used to communicate with running RDBMS

According to the needs, it is possible to configure your landscape in order to secure the different communications with SAP CC. For further information, refer to the “Securing a landscape” dedicated section available in the [SAP Convergent Charging Installation Guide](#) documentation.

Starting SAP CC

To start the Core Server instances, refer to the “Starting and Stopping the Servers” procedure described in the [SAP Convergent Charging Installation Guide](#) documentation.

Managing and Maintaining SAP CC


Managing Databases Partitioning

SAP ASE databases partitioning

The Core database tables are not partitioned due to the limitations of the Business partitioning mechanism described in the [overview section](#) of this documentation. The tables of the BART database use the storage partitioning which is implemented into SQL stored procedures dedicated to the management of these storage partitions. These SQL procedures give the possibility to:

- Create new partitions
- Purge and archive existing partitions
- Import previously archived partitions
- Clean previously imported partitions

After each operation, the record of the PARTITION_STATUS table related to the concerned partition is updated to inform that this operation successfully ended. The following table lists the possible partitions status according to the performed operation:

Operation	SQL Procedure	Status
Partition creation	MANAGE_NEW_PARTITION	1
Partition archiving	MANAGE_ARCHIVE This procedure takes into account 4 parameters: <ul style="list-style-type: none"> ■ destinationDirectory, which corresponds to file system directory in which partitions will be stored as file ■ checkCDR, which corresponds to a Boolean (0=False, 1=True) used to ensure that no unrated CDR (whose status is 0,1 or 7) exist in the CDR table before performing the purge operation ■ force, which corresponds to a Boolean (0=False, 1=True) used to stop the operation or continue with a warning ■ purgeWithNoArchive, which corresponds to a Boolean (0=False, 1=True) used to disable the archiving 	2
Partition import	IMPORT_ARCHIVE This procedure takes into account 3 parameters: <ul style="list-style-type: none"> ■ destinationDirectory, which corresponds to the file system directory containing the partitions to import ■ fromDate, the date of the first partition to import ■ numberOfDays (default 1), the number of partition to import  NOTE These dates are String values which respect a "YYYYMMDD" format	3

 **NOTE**

For further information about these SQL procedures, refer to the [SAP Convergent Charging Configuration Guide](#) documentation.


Oracle databases partitioning

The Core database tables are partitioned with the business partitioning mechanism described in the [overview](#) section of this documentation. The tables of the BART database use the storage partitioning which is implemented into SQL stored procedures dedicated to the management of these storage partitions. These SQL procedures give the possibility to:

- Create new partitions
- Purge and archive existing partitions
- Import previously archived partitions
- Clean previously imported partitions

After each operation, the record of the PARTITION_STATUS table related to the concerned partition is updated to inform that this operation successfully ended. The following table lists the possible partitions status according to the performed operation:

Operation	SQL Procedure	Status
Partition creation	MANAGE_NEW_PARTITION	1

Partition archiving	<p>MANAGE_ARCHIVE</p> <p>This procedure takes into account 3 parameters:</p> <ul style="list-style-type: none"> ■ archiveDatabase, which corresponds to the name of the database to use to transfer data ■ checkCDR, which corresponds to a Boolean (0=False, 1=True) used to ensure that no unrated CDR (whose status is 0,1 or 7) exist in the CDR table before performing the purge operation ■ force, which corresponds to a Boolean (0=False, 1=True) used to stop the operation or continue with a warning 	2
Partition purge (and cleaning)	<p>MANAGE_PURGE</p> <p>This procedure removes from the BART database all partitions of CDR table whose partition status is 2 or 6.</p>	3
Partition import	<p>IMPORT_ARCHIVE</p> <p>This procedure takes into account:</p> <ul style="list-style-type: none"> ■ 2 dates which delimit an interval used to isolate CDR table records according to the consumption date ■ The name of the database to use to transfer data <p> NOTE</p> <p>These dates are String values which respect a “YYYYMMDD” format</p>	6

 **NOTE**

For further information about these SQL procedures, refer to the [SAP Convergent Charging Configuration Guide](#) documentation.


MS SQL Server databases partitioning

The Core database tables are not partitioned due to the limitations of the Business partitioning mechanism described in the [overview section](#) of this documentation. The tables of the BART database use the storage partitioning which is implemented into SQL stored procedures dedicated to the management of these storage partitions. These SQL procedures give the possibility to:

- Create new partitions
- Purge and archive existing partitions
- Import previously archived partitions
- Clean previously imported partitions

After each operation, the record of the PARTITION_STATUS table related to the concerned partition is updated to inform that this operation successfully ended. The following table lists the possible partitions status according to the performed operation:

Operation	SQL Procedure	Status
Partition creation	MANAGE_NEW_PARTITION	1
Partition archiving	MANAGE_ARCHIVE	2

	<p>This procedure takes into account 3 parameters:</p> <ul style="list-style-type: none"> ■ archiveDatabase, which corresponds to the name of the database to use to transfer data ■ checkCDR, which corresponds to a Boolean (0=False, 1=True) used to ensure that no unrated CDR (whose status is 0,1 or 7) exist in the CDR table before performing the purge operation ■ force, which corresponds to a Boolean (0=False, 1=True) used to stop the operation or continue with a warning 	
Partition purge (and cleaning)	<p>MANAGE_PURGE</p> <p>This procedure removes from the BART database all partitions of CDR table whose partition status is 2 or 6.</p>	3
Partition import	<p>IMPORT_ARCHIVE</p> <p>This procedure takes into account:</p> <ul style="list-style-type: none"> ■ 2 dates which delimit an interval used to isolate CDR table records according to the consumption date ■ The name of the database to use to transfer data <p> NOTE</p> <p>These dates are String values which respect a “YYYYMMDD” format</p>	6

 **NOTE**

For further information about these SQL procedures, refer to the [SAP Convergent Charging Configuration Guide](#) documentation.

Synchronizing BIT Class, BIT type and Subprocess

Mechanism description

The Chargeable Items Charging process uses CIT classes to transform transactions into charged items. When SAP CC is configured to work in conjunction with an SAP ERP/FI-CA system, a mapping between the SAP ERP's BIT classes and the SAP CC's CIT classes must be defined in order to be able to generate and send billable items to the SAP ERP systems. This mapping is used by the Chargeable Items Charging process to enrich the charged items' data.

According to the configuration of the charged item file processor, a validation of these enriched data might be performed, to ensure that:

- The values of the charged item properties respect the adequate format, size, and so on
- Some properties coming from the BIT mapping are correctly filled, such as:
 - The SUBPROCESS property, which contains the name of the SAP ERP's subprocess to be associated to the billable item
 - The BITTYPE property, which contains the type of SAP ERP's billable item to create

When this validation succeeds, the charged item file processor writes these enriched data into the dedicated data files.

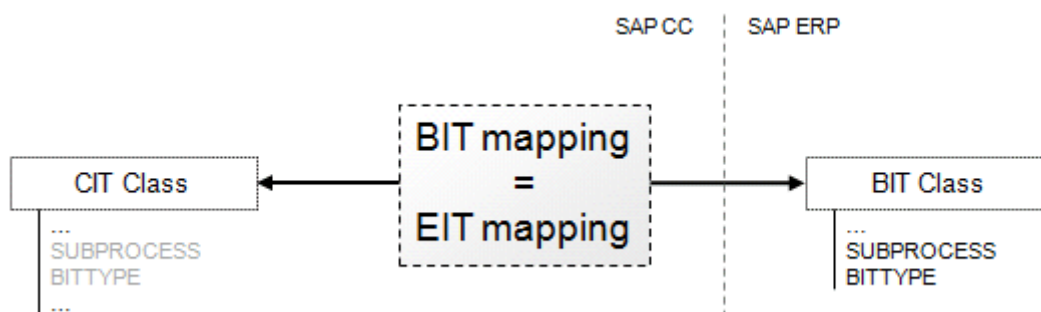
The BIT class, BIT type and subprocess synchronization mechanism gives the possibility to refresh the content of the BIT classes used into the defined BIT mappings. This ensures that no modifications have been performed on BIT classes in the SAP ERP system, which could lead to the creation of incorrect charged items rejected by the validation operation.

 **NOTE**

The default configuration of SAP CC executes a validation of the enriched charged items' data during the transaction persistency operation of the Chargeable Items Charging process. This default configuration gives the possibility to roll back the erroneous transactions before recording non-billable elements into the charged items data files. It is possible to modify this configuration to:

- Remove the validation in the transaction persistency operation, which is not recommended due to the possible erroneous elements stored into the charged items data files
- Execute the validation during the bulk loading operations, which could decrease the global loading performances and is thus not recommended
- Execute a double validation both in the transaction persistency and in the bulk loading operations, which neither necessary nor useful
- Remove any validation, which would increase the global performances but lead to heavy corrective actions in case of errors returned by the SAP ERP system

Mapping between CIT classes and BIT classes



The mapping between CIT classes and BIT classes is called:

- Billable Item Mapping for users working with the Core Tool user interface
- Exportable Item Mapping for users working with the APIs

This mapping is defined into an XML file which contains the list of properties belonging to the BIT class, and all the additional information related to these properties, such as:

- The name of the CIT class property which is mapped with each property
- The expected format of each property (type, length, and so on)
- The mandatory or optional aspect of each property
- The list of authorized values

 **EXAMPLE**

```
<exportableItemMapping externalSystemCode="CI" citcCode="SCC1" code="SCC1">
```

```

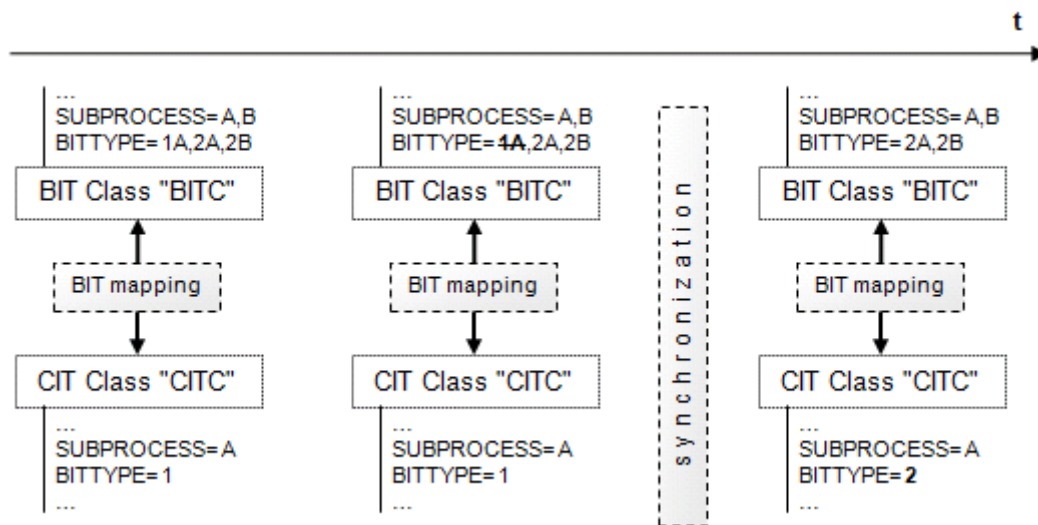
...
<exportableItemMappingField name="SRCTAID" citcFieldName="SRCTAID">
  <additionalInformation name="type" value="C" type="string"/>
  <additionalInformation name="maxLength" value="22" type="string"/>
  <additionalInformation name="decimalLength" value="0" type="string"/>
  <additionalInformation name="isMandatory" value="false" type="string"/>
</exportableItemMappingField>
<exportableItemMappingField name="SUBPROCESS" citcFieldName="SUBPROCESS">
  <additionalInformation name="type" value="C" type="string"/>
  <additionalInformation name="maxLength" value="4" type="string"/>
  <additionalInformation name="decimalLength" value="0" type="string"/>
  <additionalInformation name="isMandatory" value="false" type="string"/>
  <additionalInformation name="list" value="1000" type="string"/>
  <additionalInformation name="list" value="200" type="string"/>
  <additionalInformation name="list" value="2000" type="string"/>
</exportableItemMappingField>
<exportableItemMappingField name="BITTYPE" citcFieldName="BITTYPE">
  <additionalInformation name="type" value="C" type="string"/>
  <additionalInformation name="maxLength" value="4" type="string"/>
  <additionalInformation name="decimalLength" value="0" type="string"/>
  <additionalInformation name="isMandatory" value="false" type="string"/>
  <additionalInformation name="list" value="1000-1000" type="string"/>
  <additionalInformation name="list" value="1000-CITY" type="string"/>
  <additionalInformation name="list" value="200 3CITY" type="string"/>
  <additionalInformation name="list" value="2000-1000" type="string"/>
</exportableItemMappingField>
<exportableItemMappingField name="APPLK" citcFieldName="APPLK">
  <additionalInformation name="type" value="C" type="string"/>
  <additionalInformation name="maxLength" value="1" type="string"/>
  <additionalInformation name="decimalLength" value="0" type="string"/>
  <additionalInformation name="isMandatory" value="false" type="string"/>
</exportableItemMappingField>
...
</exportableItemMapping>

```

The CIT class which is associated to this mapping contains a superset of the BIT class properties, and all these CIT class properties can be associated and filled with information of the transactions. The only exception concerns the SUBPROCESS and BITTYPE properties of the CIT class, which can also be valued with one of the related “list” additional information defined in the mapping. To avoid errors during the validation operation, users must thus ensure that the values of the SUBPROCESS and BITTYPE properties of the CIT class always refer to an existing value of the related BIT class.

Example

The following schema describes the possible evolution of a BIT mapping in case of modification of the original BIT class on the SAP ERP side:



All the above billable item mappings link a "BITC" BIT class to a "CITC" CIT class.

The initial CITC maps:

- Its SUBPROCESS property to the "A" value, which is listed in the possible values of BITC
- Its BITTYPE property to the "1" value, which corresponds to a possible BITTYPE value of BITC

If the SAP ERP system modifies BITC and removes one of the possible BITTYPE values, then the billable item mapping possibly becomes erroneous and might lead to the generation of incorrect charged item data files or to validation errors. A manual synchronization operation gives the possibility to refresh BITC properties, and update CITC properties in order to generate correct charged items.



NOTES

- As the BITTYPE and SUBPROCESS values can be set at charge item or charge plan item level (during charged item mapping), it might also be necessary to check these elements. For further information about this feature, consult the [SAP Convergent Charging Core Tool Online Help](#) documentation
- A `java ExportableItemMappingException` exception is returned in case of erroneous during creation or modification of CIT classes or charged item mapping
- A `java BillableItemFieldSetterException` exception is returned in case of erroneous validation operation

Mechanism Limitations

- The BIT class, BIT type and subprocess synchronization mechanism is only available when SAP CC works in conjunction with the SAP ERP system
- The BIT class refreshing operation available in the BIT mapping can only be manually performed through the Core Tool user interface

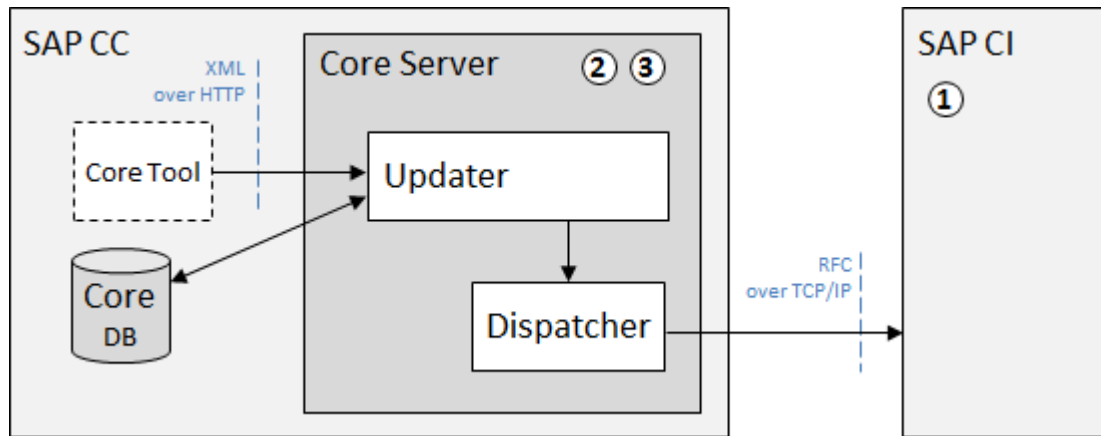
Synchronizing Chargeable Item Classes and Consumption Item Classes

This section describes how to:

- Create a consumption item mapping into SAP CC
- Synchronize chargeable items classes and consumption item classes with SAP CI

Mechanism description

The following schema shows the relationships between SAP Convergent Charging and the SAP ERP/FI-CA component of the SAP Business Suite, when creating the consumption item mappings, from consumption item classes retrieved from the SAP ERP reference system:

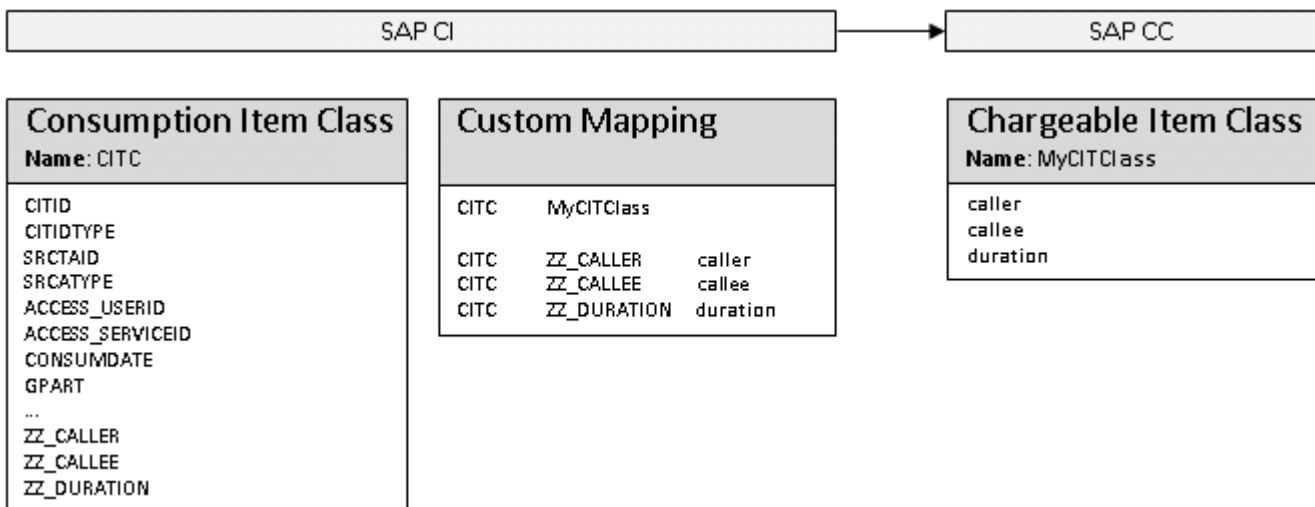


- SAP CI defines and creates the consumption item classes (1)
- SAP CC queries SAP CI in order to retrieve these consumption item classes (2) and create (3):
 - A consumption item mapping per retrieved consumption item class, required by the Bulkloaders during bulk loading operations when converting chargeable items into consumption items
 - A chargeable item class associated to the consumption item mapping and required by the Raters during charging operations when generating data files containing chargeable items

Managing consumption item classes

Consumption item classes are created by SAP CI. They basically represent chargeable item classes, enriched with additional specific fields which relate to properties of a mediation system communicating with SAP CC. This specific information relating to SAP CC can be defined into dedicated structures named customs mappings.

Each consumption item class defined in SAP CI is replicated in SAP CC as a unique chargeable item class. For more information about consumption item classes, refer to the dedicated documentation of SAP CI.



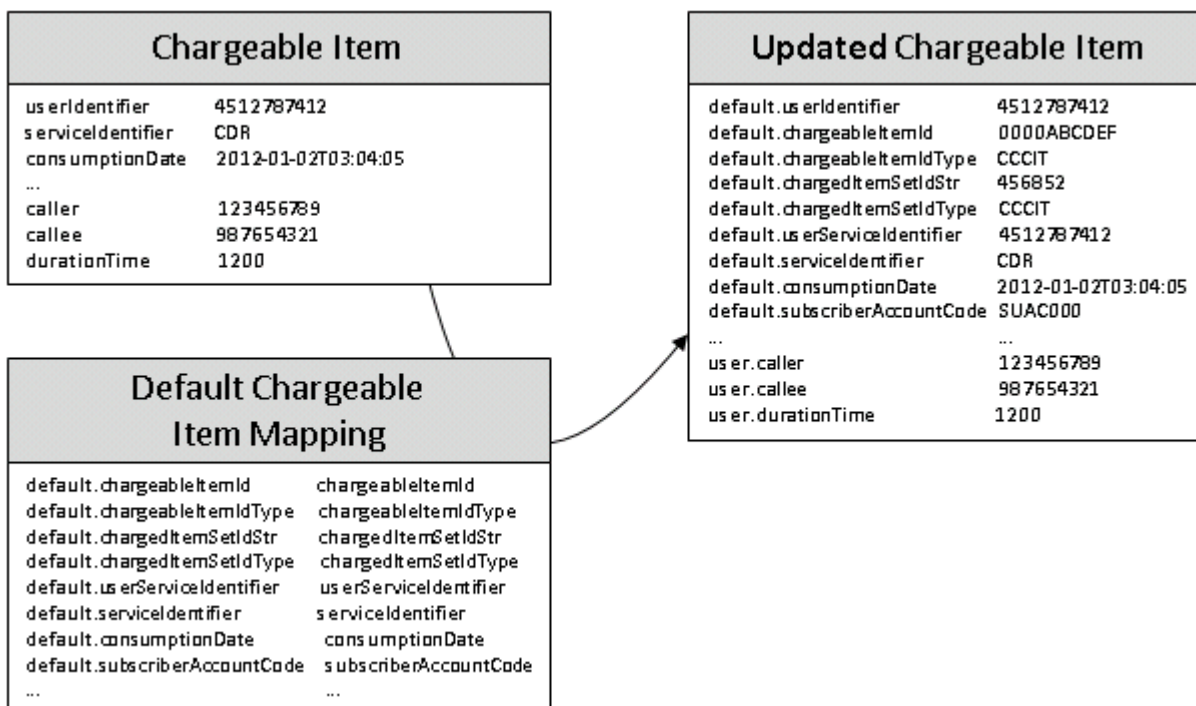
Creating consumption item mappings

To create consumption item mappings and thus synchronize consumption items classes with chargeable items classes, the following operations are performed:

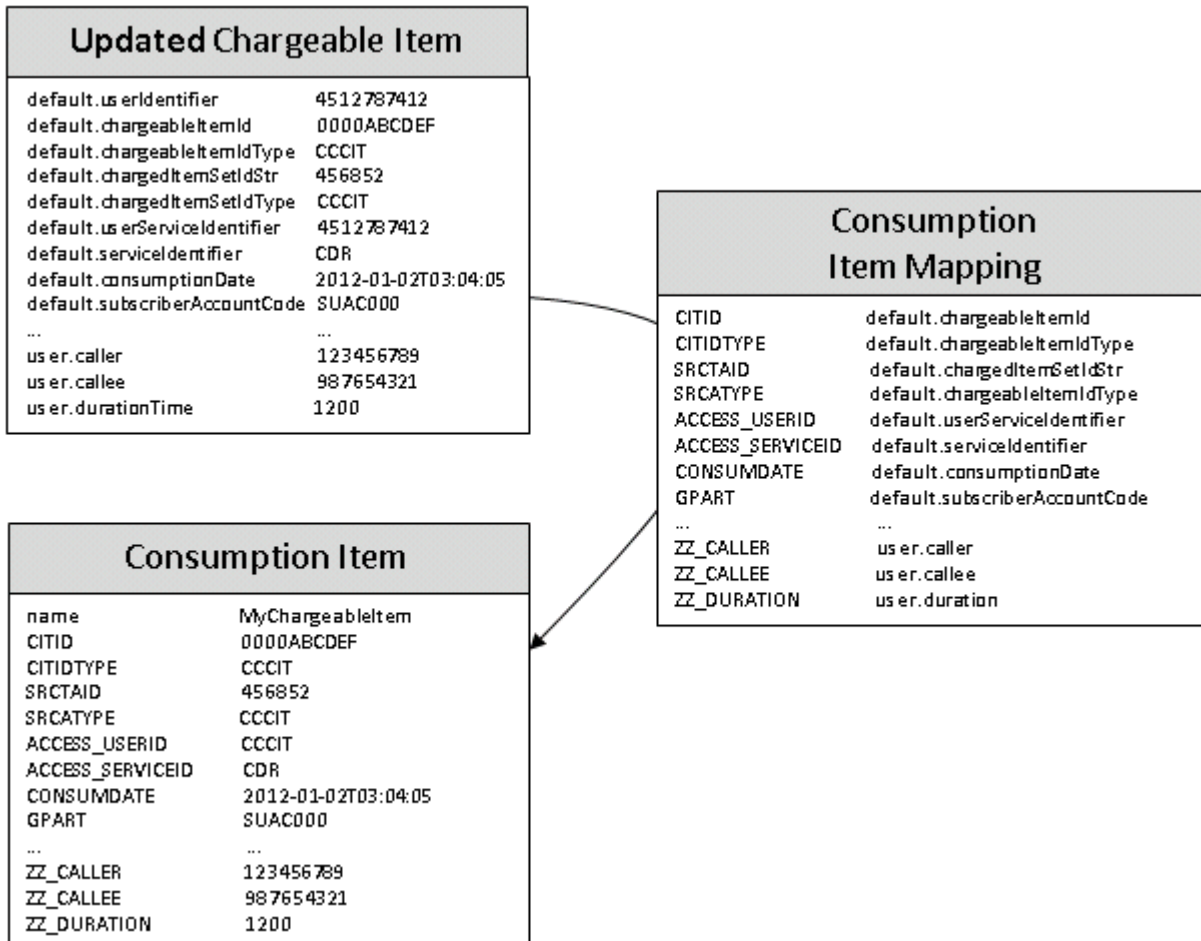
1. The Core Tool user interface retrieves the structure of a consumption item class from SAP CI
2. An initial check is performed to ensure that the consumption item mapping to create does not already exist in the Core database:
 - If the consumption item mapping to create already exists, it means that a chargeable item class is already mapped onto the retrieved consumption item class. No creation is thus performed
 - If the consumption item mapping does not exist, it is created
3. The consumption item mapping is then generated according to:
 - The default consumption item mapping configured in SAP CC, which defines the standard fields of the consumption item classes
 - The list of fields of the retrieved consumption item class, to ensure that no field is missing during future creations of consumption items
4. The updated consumption item mapping is then validated by the Core Tool user interface
5. (*Optional step*) If the chargeable item class related to the retrieved consumption item class does not exist, SAP CC creates the adequate chargeable item class and saves it into the Core database
6. The Core Tool user interface saves the consumption item mapping into the Core database

EXAMPLE

Considering a chargeable item associated to the “MyChargeableItemClass” described above, the following schema describes the use of a default chargeable item mapping, which gives the possibility to update the properties of this chargeable item (coming from a charging context) with specific default properties. The resulting chargeable item corresponds to the one which is written in data files by the Rater instance at the end of a charging operation.



The updated chargeable item is then taken into account by the Bulkloader server instance, which uses the name of the chargeable item class (part of the name of the data file containing the chargeable item to convert) to retrieve the associated consumption item mapping. Using this consumption item mapping, the Bulkloader finally creates the corresponding consumption item to send to SAP CI.



Synchronizing Currencies

Some processes or functions (such as chargeable items charging, recharging, and so on) need to compute amounts which are associated to:

- ISO currencies, which must be imported every time a modification is performed on a standard currency
- SAP currencies, which are only used in an integrated scenario with the SAP ERP/FI-CA component of the SAP Business Suite. For performance reasons, these currencies are stored in the Core database and handled as cached structures in some server instances. These SAP currencies must be synchronized every time a modification is performed in the SAP ERP/FI-CA component, in order to update both database and cached structures

For further information about the currencies import and synchronization procedures, consult:

- The “Tools/Currencies” section of the [SAP Convergent Charging Core Tool Online Help](#) documentation
- The [SAP Convergent Charging Solution Operations Guide](#) and [SAP Convergent Charging Configuration Guide](#) documentations

To audit an SAP CC Core Server system and its persistent data in your system landscape, refer to the procedures and recommendations that are available in the SAP CC product documentation and user assistance:

- [SAP Convergent Charging Solution Operations Guide](#) documentation for technical operations and tasks
- Core Tool online help documentation available in the [SAP Convergent Charging User Interfaces](#) documentation for the available procedures

Troubleshooting SAP CC

When a problem occurs during the execution of a process or function of SAP CC, it is possible to get information about the nature and/or origin of the problem, using:


- The messages provided in the console or in the graphical user interface, which contain information that can be useful to solve the problem and prevent such a situation. For further information about logging and tracing capabilities, refer to the [“Enhanced Logging and Tracing Framework”](#) section of this guide
- The information provided by the provided monitoring mechanisms, which may inform about the behavior of the different deployed components. For further information about these information, refer to the [“Monitoring SAP CC”](#) section of this guide
- The information available in the thread dump files, which provides details about the execution context of the different threads that are executed by the instances of the Core Server system. For further information about these information, refer to the description of the [“Troubleshooting \(Advanced\): Thread Dump Generation”](#) group available in the [SAP Convergent Charging System Parameters Reference](#) documentation

For global information about the troubleshooting procedures, consult the [“Troubleshooting”](#) section of the [SAP Convergent Charging Solution Operations Guide](#) documentation.






Upgrading SAP CC

To upgrade an SAP Convergent Charging system landscape from a replaced 4.0 release to a 5.0 release, consult the [SAP Convergent Charging Upgrade Guide](#) documentation.

Typographic Conventions

Type Style	Description
	Arrows separating the parts of a navigation path, for example, menu options.
<i>Example Text</i>	Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Cross-references to other documentation
Example text	Emphasized words or phrases in body text, graphic titles, and table titles
EXAMPLE TEXT	Technical names of system objects. These include report names, program names, transaction codes, table names, and key concepts of a programming language when they are surrounded by body text, for example, SELECT and INCLUDE.
Example text	Output on the screen. This includes file and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.
<code>Example text</code>	Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.
<Example text>	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system, for example, “Enter your <User name>”.
<code>EXAMPLE</code>	Keys on the keyboard, for example, <code>F2</code> or <code>ENTER</code> .

Icons

Icon	Meaning
	Caution
	Example
	Note
	Recommendation
	Syntax

Copyrights

© Copyright 2017 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company. The information contained herein may be changed without prior notice.

Some software products marketed by SAP SE or an SAP affiliate company and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Excel, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, System z9, z10, z9, iSeries, pSeries, xSeries, zSeries, eServer, z/VM, z/OS, i5/OS, S/390, OS/390, OS/400, AS/400, S/390 Parallel Enterprise Server, PowerVM, Power Architecture, POWER6+, POWER6, POWER5+, POWER5, POWER, OpenPower, PowerPC, BatchPipes, BladeCenter, System Storage, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, Parallel Sysplex, MVS/ESA, AIX, Intelligent Miner, WebSphere, Netfinity, Tivoli and Informix are trademarks or registered trademarks of IBM Corporation.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP Business ByDesign, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE or an SAP affiliate company in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects S.A. in the United States and in other countries. Business Objects is an SAP company.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP SE or an SAP affiliate company and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

Disclaimer

Some components of this product are based on Java™. Any code change in these components may cause unpredictable and severe malfunctions and is therefore expressly prohibited, as is any decompilation of these components.

Any Java™ Source Code delivered with this product is only to be used by SAP's Support Services and may not be modified altered in any way.

SAP SE
Dietmar-Hopp-Allee 16
69190 Walldorf
Germany
T +49/18 05/34 34 34
F +49/18 05/34 34 20
www.sap.com