CID Reference Guide

V4.0
Document ID: TSGN-06-4.0-01
Date Published: 9.10.03

edocs, Inc., One Apple Hill Drive, Suite 301, Natick, MA 01760

Preface

In This Section

Using this Manual iv
Finding the Information You Need vii
If You Need Help ix

Using this Manual

Welcome to the Customer Interaction Datastore (CID) Reference Guide.

This manual is a reference guide to the CID and its components.

Before You Get Started

You should be familiar with the following:

- Your application
- Designing or working with databases

Who Should Read this Manual

This manual is a reference guide for the Customer Interaction Datastore (CID). This guide is for anybody who needs information about content and structure of the CID.

The CID Reference Guide deals with the conceptual model of the CID and its submodels. For more detailed information about the CID, refer to the *CID Reference* documentation corresponding to your database platform.

- Administrators

  You will find information about the conceptual structure of the CID. There is also some information about customizing that is important when working with solutions with customizations.

- Developers

  This guide contains the conceptual structure of the CID you can use to understand how the CID organizes information. You need to ba familiar with the tables and their relationships. When developing your solution, you need to install the sample data in order to use the Presentation Manager channels.

- Project Architect

  The most useful part of this document is the organization of the tables in the CID submodels. You use this guide to understand the content of the submodels and how the CID manages the information inside them. You also need to be aware of the restrictions and limitations to customizing the CID.

- Project Manager

  As this reference guide contains information about the CID and its submodels, you will find almost all of the information in this guide as useful when managing your project.

How this Manual is Organized

This manual covers the following:

- **Overview of the CID**

  This chapter covers the basics of the Customer Interaction Datastore (CID) and its role in Account Management solutions.

  It contains information about:

  - The CID
  - How the CID works with other components

- **Database Submodels**

  This chapter covers the submodels of the CID.

  For each submodel, it contains:

  - A diagram of the tables and their relationships
  - Description of the submodel
  - Additional information required when working with the submodel

- **Sample Data**

  This chapter covers working with the sample data of the CID.

  It contains information about:

  - Loading the sample data
  - Sample reference data
  - Sample service data
  - Sample user data

- **Customizing the CID**

  This chapter covers the basic rules when customizing the CID.

  It contains information about:

  - Reserved tables
  - Reserved ranges
  - Modifying tables and records

What Typographical Changes and Symbols Mean

This manual uses the following conventions:

TYPEFACE	MEANING	EXAMPLE
Italics	Manuals, topics or other important items	Refer to *Developing Connectors*.
Small Capitals	Software and Component names	Your application uses a database called the CID.
Fixed Width	File names, commands, paths, and on screen commands	Go to `//home/my file`

Finding the Information You Need

The product suite comes with comprehensive documentation set that covers all aspects of building Account Management solutions. You should always read the release bulletin for late-breaking information.

Getting Started

If you are new to the Edocs Telco Solutions, you should start by reading *Introducing Telco Service Manager*. This manual contains an overview of the various components along with a list of the available features. It introduces various concepts and components you must be familiar with before moving on to more specific documentation. Once you have finished, you can read the manual which covers different aspects of working with the application. At the beginning of each manual, you will find an introductory chapter which covers concepts and tasks.

Designing Your Solution

While reading *Introducing Telco Service Manager*, you should think about how the different components can address your Account Management Solution's needs.

You can refer to *Developing Telco Service Manager* for information about extending the object model, application security, and other design issues. The *CID Reference Guide* also gives you the information about how the information in your solution is managed and stored.

Installing Your Telco Service Manager

You should start by reading the Release Bulletin. For detailed installation and configuring information, refer to *Installing Telco Service Manager*. This manual covers installing TSM on one or more computers. It also contains the information you need to configure the different components you install. You might also refer to *Developing Telco Service Manager* and *Developing Connectors for Telco Service Manager* as these manuals contain information on customizing applications and working with other software.

Building Account Management Solutions

If you are designing and programming *Telco Service Manager*, you have several different sources of information. If you are programming the user interface of the solution, you should read *Developing User Interfaces for Telco Service Manager*. You also refer to the BLM Specification for detailed information about programming the user interface. For configuring the various components, you refer to *Installing Telco Service Manager*and sections in other documents which deal with the component to configure.

If you are working with the business logic of your solution, you should read *Developing Telco Service Manager*. You can also refer to the *BLM Reference Guide* for more information about the design and structure of the BLM object model. For information about how this information is stored, you should refer to the *CID Reference Guide* along with the CID Reference documentation for your database. In order to develop your application, you most likely will need to install and run the Loopback Connector. This component mimics back-end applications for development purposes. For information about installing and running this component, refer to *Using the Loopback Connector with Telco Service Manager*.

Integrating Account Management Solutions

If you are involved in configuring your solution to work with Operation Support Software (OSS), you should read *Developing Connectors with Telco Service Manager*. This manual helps you understand the integration architecture and shows you how to build connectors to connect to today's market-leading OSS software. You can also read *Using the Loopback Connector with Telco Service Manager* for information about a connector built for development purposes. Other manuals you can refer to for information about configuring your application include *Introducing Telco Service Manager* and *Developing Telco Service Manager*.

Managing Telco Service Manager (TSM)

If you are responsible for managing TSM, you should read the *Installing Telco Service Manager* for information about configuring various components and information about working with different application servers. *Administrating Telco Service Manager* covers what you need to know about managing your solution at runtime. For information about OSS systems, you should read *Developing Connectors with Telco Service Manager*.

If You Need Help

Technical support is available to customers who have valid maintenance and support contracts with edocs. Technical support engineers can help you install, configure, and maintain your edocs application.

To reach the U.S. Service Center, located in Natick, MA (Monday through Friday 8:00am to 8:00pm EST):

- Telephone: 508.652.8400
- Toll Free: 877.336.3362
- E-support: support.edocs.com (This requires a one-time online registration)
- E-mail: support@edocs.com

When you report a problem, please be prepared to provide us the following information:

- What is your name and role in your organization?
- What is your company's name?
- What is your phone number and best times to call you?
- What is your e-mail address?
- In which edocs product did a problem occur?
- What is your Operating System version?
- What were you doing when the problem occurred?
- How did the system respond to the error?
- If the system generated a screen message, please send us that screen message.
- If the system wrote information to a log file, please send us that log file.

If the system crashed or hung, please tell us.

Contents

Preface iii

Overview of the CID 15

About the CID 16
About the CID and TSM 17

Database Submodels 19

About the Database Submodels 20
ACTION 21
Submodel Diagram 21
About this Submodel 22
AUDIT 26
Submodel Diagram 26
About this Submodel 27
Additional Details 27
BILLING ACCOUNT 30
Submodel Diagram 30
About this Submodel 31
PAYMENT_INFO Table 31
COMMERCIAL OFFER 32
Submodel Diagram 32
About this Submodel 33
CONTACT 34
Submodel Diagram 34
About this Submodel 35
CONTRACT 36
Submodel Diagram 36
About this Submodel 38
DOCUMENTATION 39
Submodel Diagram 39
About this Submodel 39
FILTER 40
Submodel Diagram 40
About this Submodel 41
INVOICE 42
Submodel Diagram 42
About this Package 43
NOTIFICATION 44
Submodel Diagram 44
About this Submodel 45
ORGANIZATION 46
Submodel Diagram 46
About this Submodel 47

ORGANIZATION VIEW 49
Submodel Diagram 49
About this Submodel 50
PARAMETER 51
Submodel Diagram 51
About this Submodel 52
PAYMENT 58
Submodel Diagram 58
About this Submodel 58
PAYMENT METHOD 59
Submodel Diagram 59
About this Submodel 59
PROCUREMENT 60
Submodel Diagram 60
About this Submodel 61
RATEPLAN 62
Submodel Diagram 62
About this Submodel 63
REQUEST 65
Submodel Diagram 65
About this Submodel 66
REQUEST_STATUS 66
SERVICE 68
Submodel Diagram 68
About this Submodel 69
SHOPPING CART 70
Submodel Diagram 70
About this Submodel 71
STRING LOCALIZATION 72
Submodel Diagram 72
About this Submodel 72
SYSTEM 73
Submodel Diagram 73
About this Submodel 73
OBJECT_TYPE Table 74
TROUBLE TICKET 75
Submodel Diagram 76
About this Submodel 77
USAGE INFORMATION 78
Submodel Diagram 78
About this Submodel 79
USER 80
Submodel Diagram 80
About this Package 82
Additional Details 82
Personalization Data 83

Sample Data 85

About the Sample Data 86
Creating the CID Database 87
Contents of the Sample Data 89
Reference Data 89
Product Catalog Data 89
User and Associated Customer/Dealer/CSR Information 89

Customizing the CID 91

About Customizing the CID 92
Reserved Tables 93
Modifying Tables and Records 94

Index 97

CHAPTER 1

Overview of the CID

In This Section

About the CID 16
About the CID and TSM..... 17

CHAPTER 2

About the CID

The Customer Interaction Datastore (CID) contains a subset of data from the existing OSS system that is necessary for your Account Management solution

The CID comprises the following sections:

- Customer / dealer / data
- Request queue management
- Notification queue management
- Product catalog (rate plans, services...)
- User data

The purpose of the CID is to facilitate the delivery of data required by TSM without directly accessing the OSS system. The advantages of the CID are:

- Autonomy from the legacy system
- Scalability
- Availability of data

About the CID and TSM

The Customer Interaction Datastore (CID) is a highly normalized relational database model, designed to reflect the needs and structure of the modern communications service provider. The database schema is constructed around an online transactional processing (OLTP) business model focused on the sale, delivery, support and management of communications products and services to business and consumer markets.

The CID is the reference repository for all data specific to customer self-service, such as a user's preferences for the presentation and formatting of the self-service portal, and descriptive text and images for the service catalog. In addition, the CID creates a data cache, holding data copied from the multiple back-office systems and stored locally. This asynchronous caching approach decouples self-service performance from back-office systems, improving responsiveness as well as protecting the service provider's mission-critical BSS/OSS and particularly billing systems from high volume Internet traffic.

The CID also has a component used to allow the approval of changes to information. This component is called the approval sequencer which is responsible for obtaining the required permission before submitting the changes for processing.

The CID includes:

- The Customer Interaction Datastore
- Set of administration tools
- Sample reference and customer data sets
- Approval Sequencer

For more information about:

- The structure and organization of the CID, refer to the *CID Reference Documentation* corresponding to your database.

- Using the administration tools, refer to *Administrating Telco Service Manager*.

- Working with the Approval Sequencer, refer to *Working with Approvals* in *Developing Telco Service Manager*.

CHAPTER 3

Database Submodels

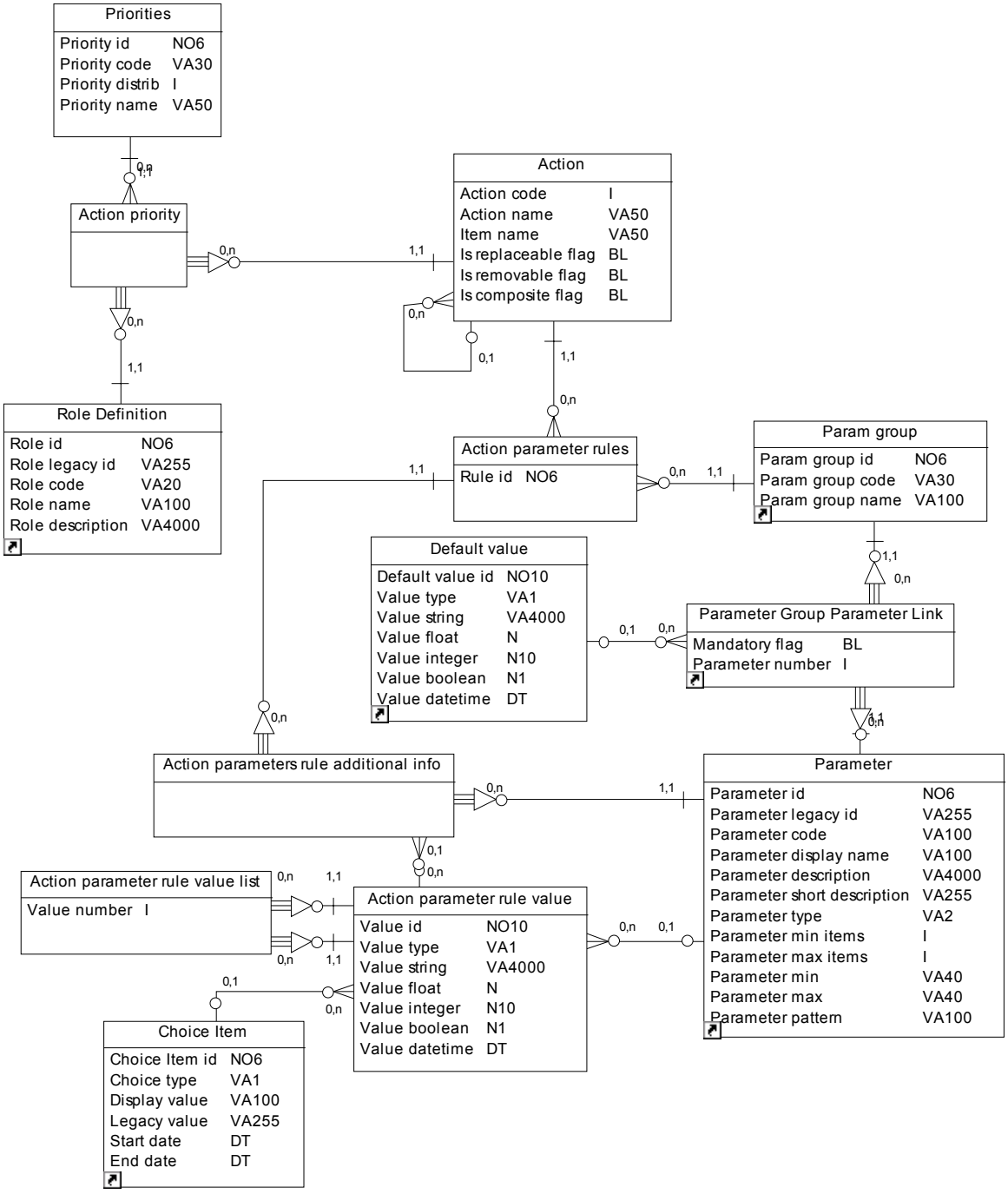
In This Section

About the Database Submodels 20
ACTION 21
AUDIT 26
BILLING ACCOUNT 30
COMMERCIAL OFFER 32
CONTACT 34
CONTRACT 36
DOCUMENTATION 39
FILTER 40
INVOICE 42
NOTIFICATION 44
ORGANIZATION 46
ORGANIZATION VIEW 49
PARAMETER 51
PAYMENT 58
PAYMENT METHOD 59
PROCUREMENT 60
RATEPLAN 62
REQUEST 65
SERVICE 68
SHOPPING CART 70
STRING LOCALIZATION 72
SYSTEM 73
TROUBLE TICKET 75
USAGE INFORMATION 78
USER 80

About the Database Submodels

ACTION

Submodel Diagram



About this Submodel

This submodel defines actions. An action can be:

- An elementary action:
  - Add customer
  - Add contract
  - Add services
  - and so on
- A composite action (a set of elementary actions submitted together).

  For example, the Register New Customer action is composed of the following actions:
  - Add customer
  - Add contract
  - Add services

The action name specifies the display name of a request in the request queue. The action item name specifies the item name of a shopping cart item.

A composite action can generate a composite request and one elementary request per call to an elementary execution action (if the elementary execution action generates a request).

Action Package Flags

The Action submodel contains the following flags:

- `IS_REPLACEABLE`: indicates whether a new item of the same type and related to the same object as a previous item in the product cart must (by default) replace it or create a new entry
- `IS_REMOVABLE`: indicates whether an action item of this type can be removed from the product cart.

This table contains information on the different action types and their default values.

ACTION TYPE	ACTION ITEM	IS REPLACEABLE	IS REMOVABLE
Add billing account	Modified organization	True	False
Add billing contact	Billing contact	True	False
Add contract	New contract	False	True
Add legal contact	Legal contact	True	False
Add level	Level	False	True

ACTION TYPE	ACTION ITEM	IS REPLACEABLE	IS REMOVABLE
Add login	Login	True	True
Add manager	Manager	False	True
Add member	Member	False	True
Add organization	New organization	False	False
Add organization view		False	False
Add organization view level		False	False
Add service	New service	False	True
Associate dedicated offer	Dedicated offer	False	True
Change contract owner	Contract owner	True	True
Change language	New language	True	True
Change password	Changed password	True	True
Change rate plan	New contract rate plan	True	True
Change services		False	False
Create contract		False	False
Create level		False	False
Create member		False	False
Create order		False	False
Create organization		False	False
Create trouble ticket		False	False
Declare loss or theft		False	False
Declare/undeclare payment responsible	Payment responsible	True	False
Dissociate dedicated offer	Dissociated dedicated offer	False	True
Login		False	False
Logout		False	False
Migrate contract		False	False
Migrate contract	Contract migration	True	True
Modify billing account	Modified billing account	True	True
Modify billing contact	Billing contact modification	True	True

ACTION TYPE	ACTION ITEM	IS REPLACEABLE	IS REMOVABLE
Modify contract	Modified contract	True	True
Modify contract status	New contract status	True	True
Modify legal contact	Legal contact modification	True	True
Modify level	Modified level	True	True
Modify line	Modified line	True	True
Modify member	Modified member	True	True
Modify organization	Modified organization	True	True
Modify organization view		False	False
Modify payment information	Modified payment method	True	False
Modify service	Modified service	True	False
Modify trouble ticket		False	False
Notify	Request	False	False
Order	Order	False	False
Order documentation		False	False
Recharge prepaid		False	False
Remove service	Service removed	False	False
Replace service	Service replaced	True	False
Request OSS approval	Request	False	False
Set personalization data		True	False

Additional Parameters for a Request

You can add additional parameters to an elementary action to customize it.

To add additional parameters

1 Define the rules in the `ACTION_PARAM_RULE_ADD_INFO` table

Choose the rule ID (`ACTION_PARAM_RULE_ID` column) and define the values of `PARAM_ID` in the:

- `ACTION_PARAM_RULE_VALUE` and

- ACTION_PARAM_RULE_VALUE_LIST tables

  For example: the rule can be `contract type (CO_TYPE_ID) = 4`.

  For more information about storing a parameter value, refer to the *PARAMETER* section in this manual.

The rule can contain several lines if it depends on the value of several parameters

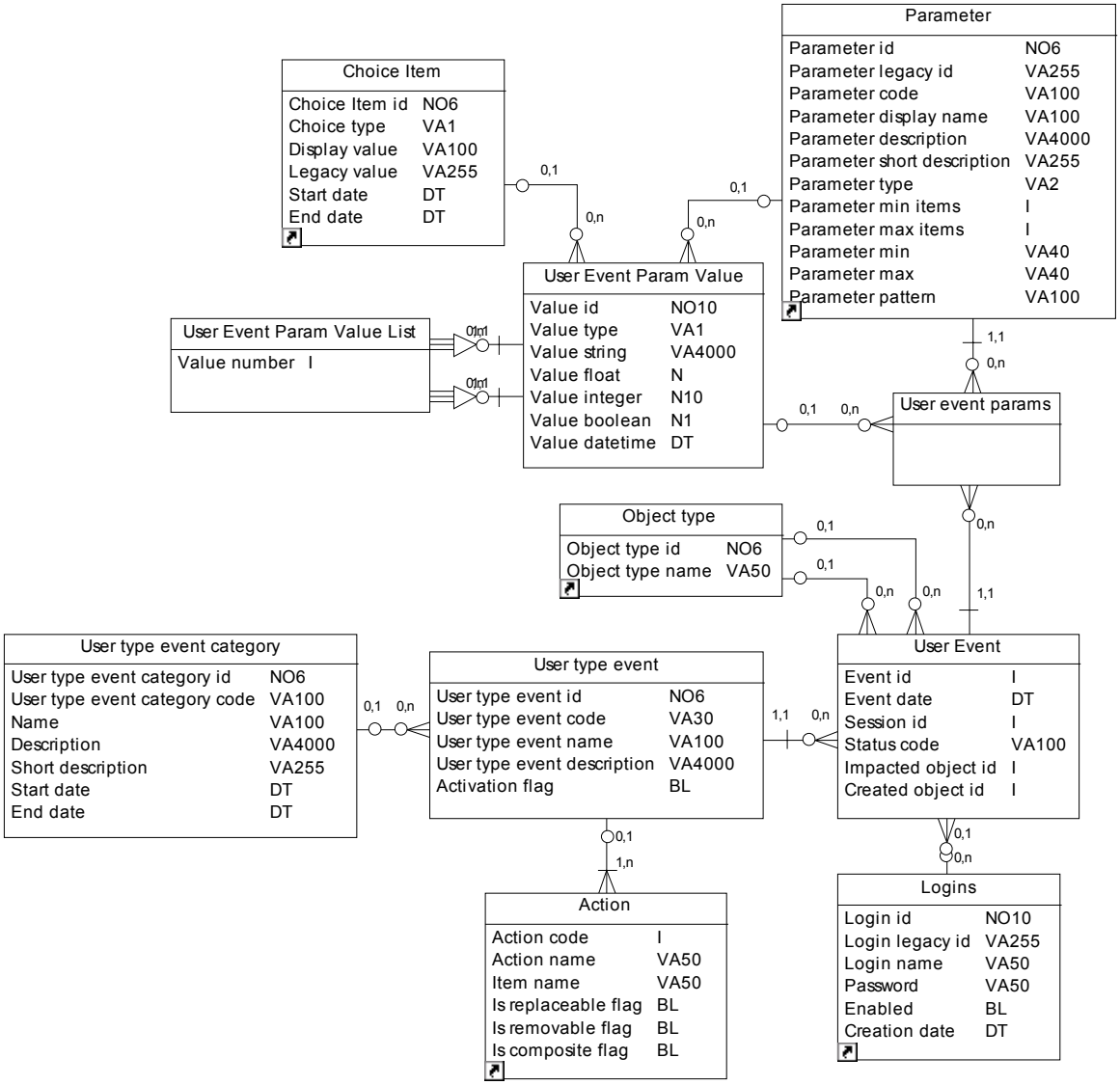
2 Define the parameter group in the PARAM_GROUP table:

Choose the ID and name the group. Define the set of parameters in the PARAM_GROUP_PARAM_LINK table.

3 Associate a parameter group for a given rule to a request type in the ACTION_PARAM_RULES table.

AUDIT

Submodel Diagram



About this Submodel

This submodel defines user events. A user event represents an action performed by a user or an event sent by an OSS which is then processed.

User events belong to one of the following categories:

- Session events (login, logout, session expiration)
- Execution features
- Custom user events
- DO events generated by an OSS

Each Execution feature has a related User Event Type.

A user event is defined as follows:

A user event:

- Is related to the user who generated it (null for the Synchronizer connector)
- Has an associated type (User Event Type located in the `USER_TYPE_EVENT` table)
- May have some parameter values giving details on the user event (DO events have standard parameters)
- May be related to created or impacted objects

For an execution feature event, the parameters are the same as the parameters of the related request. For more information, refer to the `REQUEST` submodel in this manual.

The login user event has the following parameters:

- Application server session ID
- Access channel
- HTTP user agent string

Additional Details

DO events have the following parameters:

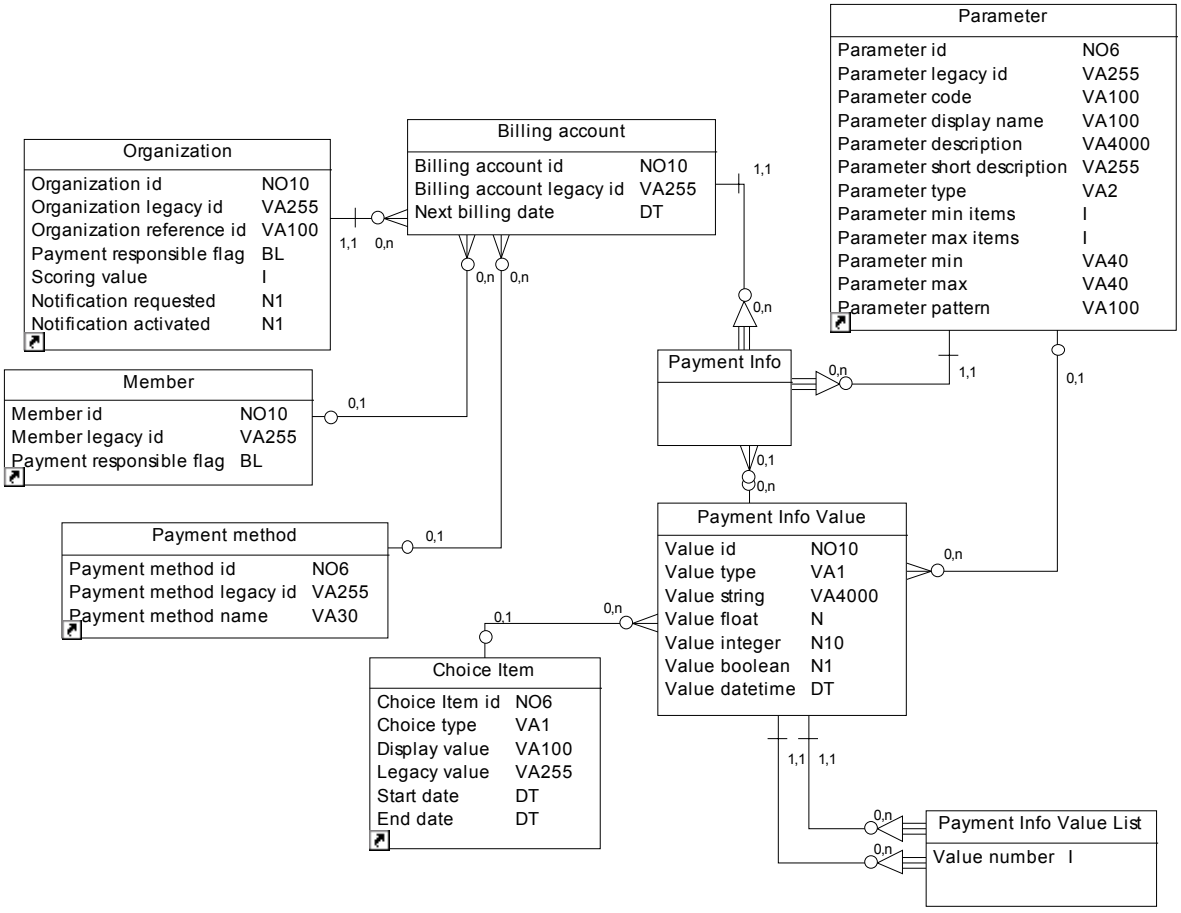
DO REQUEST	EVENT TYPE	IMPACTED OBJECT	CREATED OBJECT	EVENT PARAMETERS
Do add organization	Organization added	Null	Created Organization	Scoring + Organization additional info

DO REQUEST	EVENT TYPE	IMPACTED OBJECT	CREATED OBJECT	EVENT PARAMETERS
Do modify organization	Organization modified	Modified organization	Null	Scoring + Organization additional info
Do add level	Level added	Organization or level	Created level	Level additional info
Do add member	Member added	Organization or level	Created Member	Member additional info
Do modify member	Member modified	Modified Member	Null	Member additional info
Do add contract	Contract added	Organization or level	Created contract	Contract type Contract status Line number Rate plan Contract additional info
Do change contract status	Contract status modified	Contract	Null	Contract status
Do migrate contract	Contract migrated	Organization or level if contract created else contract id	Contract Id if one created	Contract type Contract status Line number Rate plan Contract additional info
Do add contact	Legal Contact added (if contact role is legal)	Organization, level or member	Null	Contact attributes (+ contact additional info)
	Billing contact added (if contact role is billing)	Organization, level or member	Null	Contact attributes (+ contact additional info)
Do modify contact	Legal Contact modified (if contact role is legal)	Organization, level or member	Null	Contact attributes (+ contact additional info)
	Billing contact modified (if contact role is billing)	Organization, level or member	Null	Contact attributes (+ contact additional info)
Do add login	Login added	Member	Null	Login value Login roles
Do add billing account	Billing account added	Organization, level or member	Billing account id	Payment method Payment method parameters
Do modify payment information	Payment Information modified	Organization, level or member	Null	Payment method Payment method parameters

DO REQUEST	EVENT TYPE	IMPACTED OBJECT	CREATED OBJECT	EVENT PARAMETERS
Do add service	Service added	Contract	Null	Service Service parameters values
Do modify service parameters	Service modified	Contract	Null	Service Service parameters values
Do remove service	Service removed	Contract	Null	Service
Do change rate plan	Rate plan changed	Contract	Null	Rate plan
Do add dedicated offer	New dedicated offer	Organization	Null	Dedicated offer
Do remove dedicated offer	Removed dedicated offer	Organization	Null	Dedicated offer
Do declare payment responsible	Change payment responsible	Organization, level or member	Null	Flag Is payment responsible
Do modify level	Level modified	Level	Null	Additional information
Do change contract owner	New Owner	Contract	Null	Removed owner New owner
Do modify billing account	Billing account modified	Organization, level or member	Null	Additional information
Do modify contract	Contract modified	Contract	Null	Contract level + Additional information
Do modify Line	Line modified	Contract	Null	Line number + Additional information

BILLING ACCOUNT

Submodel Diagram



About this Submodel

This submodel defines billing accounts. A billing account defines a set of contracts on one invoice. Each billing account has an organization or member which is responsible for payment. A billing account has one responsible for payment.

The link between contracts and a billing account is implicit. All contracts linked to, or below the responsible for payment on the billing account are implicitly related to it.

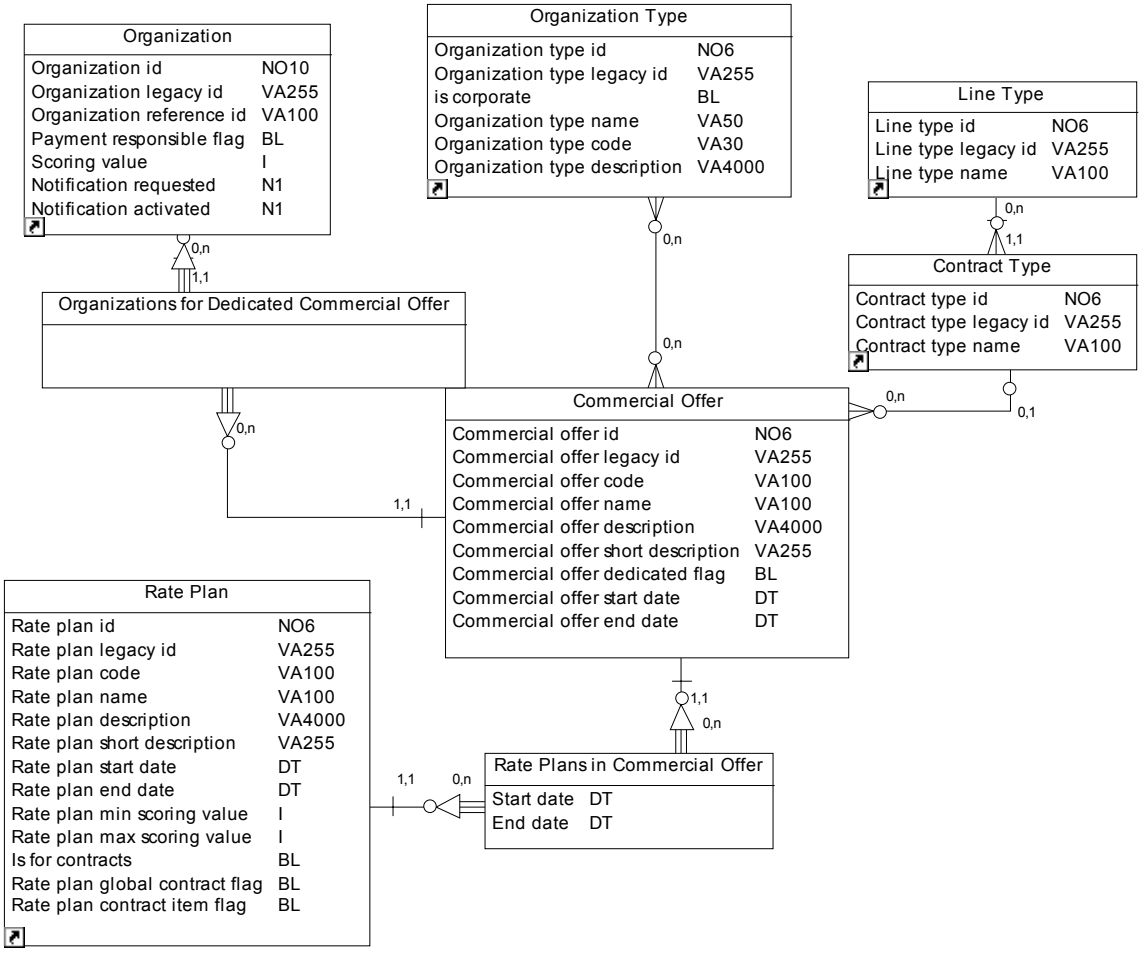
A payment method (with related information) can be assigned to a billing account.

PAYMENT_INFO Table

This table contains the payment information for a billing account. It defines the values of each parameter of the payment method that the billing account has.

COMMERCIAL OFFER

Submodel Diagram



About this Submodel

This submodel defines commercial offers. A commercial offer is a set of rate plans and associated products. It is dedicated to one contract type. The organization type-commercial offer link defines which commercial offers are available for which organization types.

The availability of an offer depends on:

- The organization type of the buyer; an offer can be available for more than one organization type. (If no organization type is defined, this means the offer is available for all organization types)
- A specific contract-type-to-segment offer by a telecom operator product line (prepaid mobile, postpaid mobile, fixed line, data…).

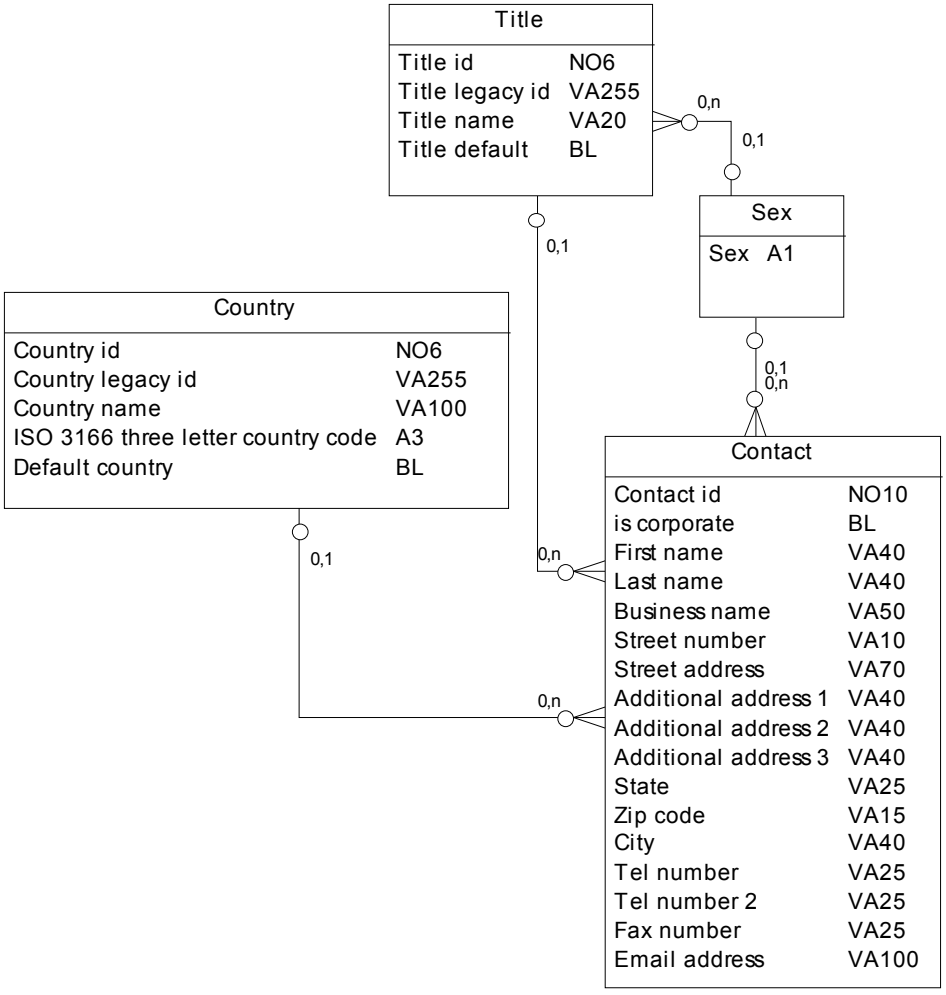
  This criterion is optional. Some offers can be defined for services not related to a contract (for example, loyalty offer).

A commercial offer can also be dedicated to one or more companies (for example, a main organization and its subsidiaries).

To prevent an unauthorized user from viewing a dedicated offer which is not yet linked to an organization, the offer class has an IS DEDICATED attribute.

CONTACT

Submodel Diagram



About this Submodel

This submodel defines contacts. A contact defines the way to contact an individual, a company, or a company level.

An individual contact is defined by:

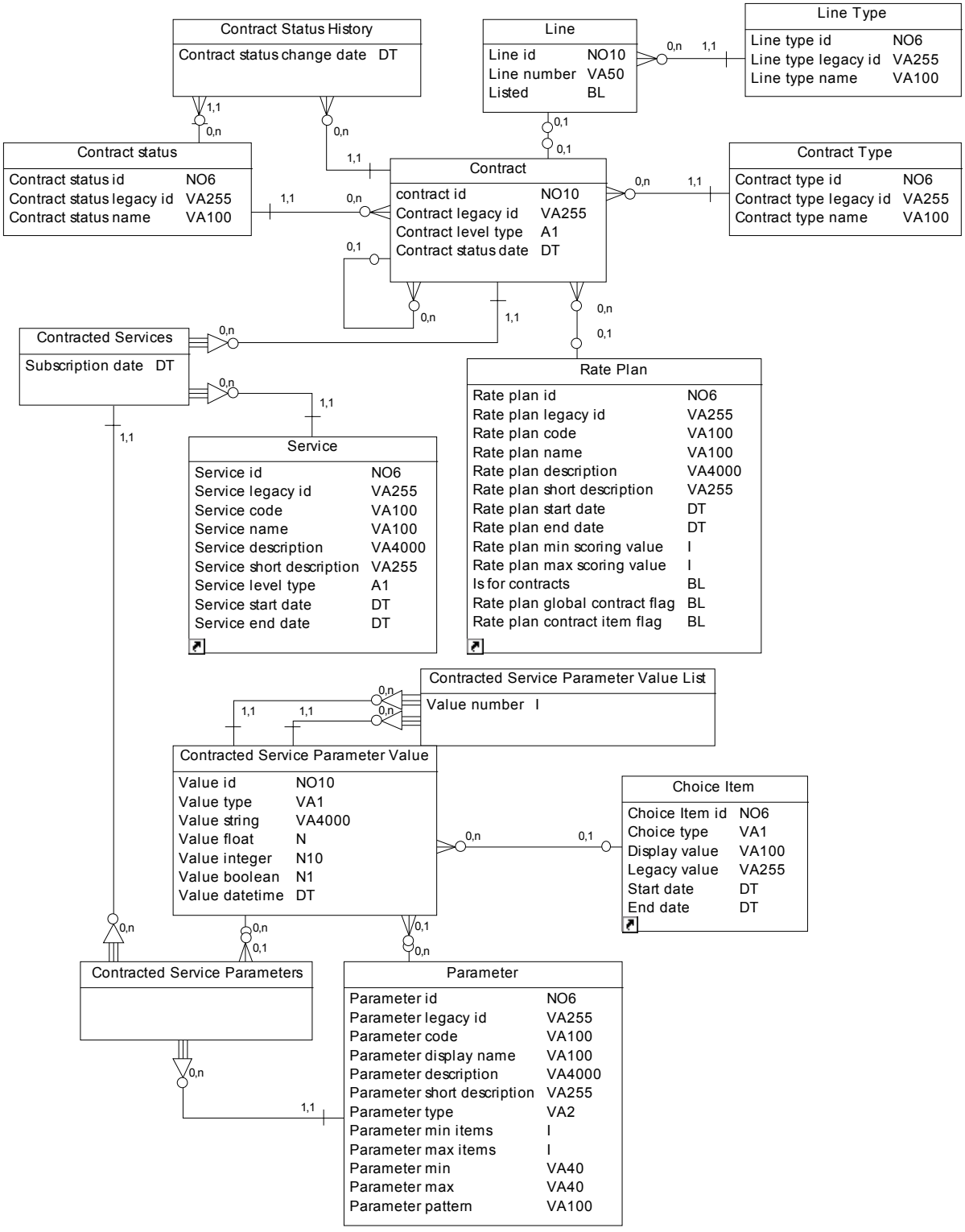
- Identity (first name, last name, sex, title)
- Address, phone numbers, e-mail address
- A business contact (Company or Company level) is defined by:
- Business name (Company name or level name)
- Address, phone numbers, e-mail address

All of these elements are optional (for example: a contact can be defined just by his or her e-mail address).

The `IS_CORPORATE` flag specifies whether a contact is an individual or corporate contact.

CONTRACT

Submodel Diagram


Contract Status History
Contract status change date DT
Line
Line id NO10
Line number VA50
Listed BL
Line Type
Line type id NO6
Line type legacy id VA255
Line type name VA100
Contract status
Contract status id NO6
Contract status legacy id VA255
Contract status name VA100
Contract
contract id NO10
Contract legacy id VA255
Contract level type A1
Contract status date DT
Contract Type
Contract type id NO6
Contract type legacy id VA255
Contract type name VA100
Contracted Services
Subscription date DT
Service
Service id NO6
Service legacy id VA255
Service code VA100
Service name VA100
Service description VA4000
Service short description VA255
Service level type A1
Service start date DT
Service end date DT
Rate Plan
Rate plan id NO6
Rate plan legacy id VA255
Rate plan code VA100
Rate plan name VA100
Rate plan description VA4000
Rate plan short description VA255
Rate plan start date DT
Rate plan end date DT
Rate plan min scoring value I
Rate plan max scoring value I
Is for contracts BL
Rate plan global contract flag BL
Rate plan contract item flag BL
Contracted Service Parameter Value List
Value number I
Contracted Service Parameter Value
Value id NO10
Value type VA1
Value string VA4000
Value float N
Value integer N10
Value boolean N1
Value datetime DT
Choice Item
Choice Item id NO6
Choice type VA1
Display value VA100
Legacy value VA255
Start date DT
End date DT
Contracted Service Parameters
Parameter
Parameter id NO6
Parameter legacy id VA255
Parameter code VA100
Parameter display name VA100
Parameter description VA4000
Parameter short description VA255
Parameter type VA2
Parameter min items I
Parameter max items I
Parameter min VA40
Parameter max VA40
Parameter pattern VA100

About this Submodel

This submodel defines contracts. A contract item describes the basic service subscribed by a customer.

A contract can be:

- a global contract
- a contract item

A global contract is a set of contract items, but can also be related to a specific line (the primary line). A global contract can be located under a different hierarchy node from the contract items it is related to.

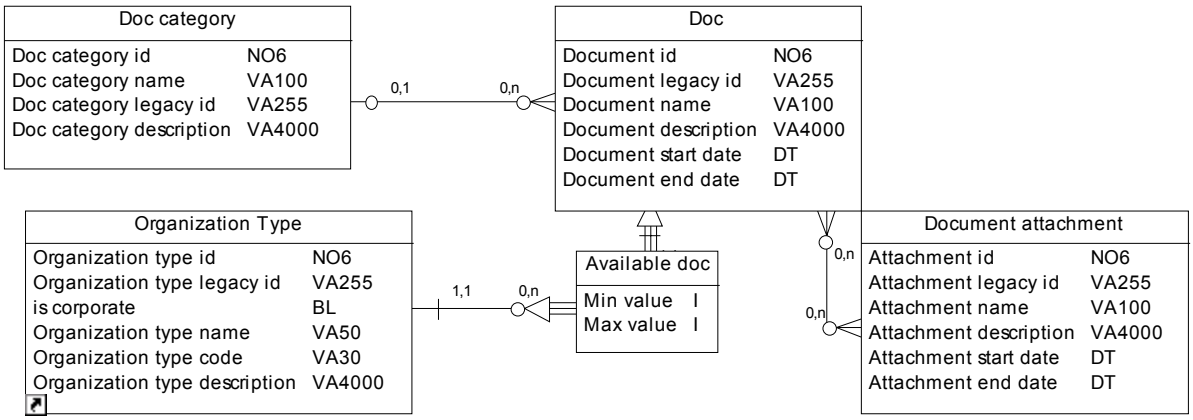
For the current version, a global contract **must** be related to a line. Global contracts and contract items are managed with no link between them.

The line type defines the basic network service type (GSM, fixed, data...). The contract type defines the specific management policy of the contract (GSM postpaid, GSM prepaid), and must be consistent with the line type.

A contract can have different statuses, but only one status at a time. `CONTRACT_STATUS_HISTORY` holds the status history of a contract.

DOCUMENTATION

Submodel Diagram



About this Submodel

This submodel defines documentation. Documentation is a piece of documentation with one or more optional enclosures that can be ordered with it. These pieces of documentation can be grouped by category.

Documentation is defined by:

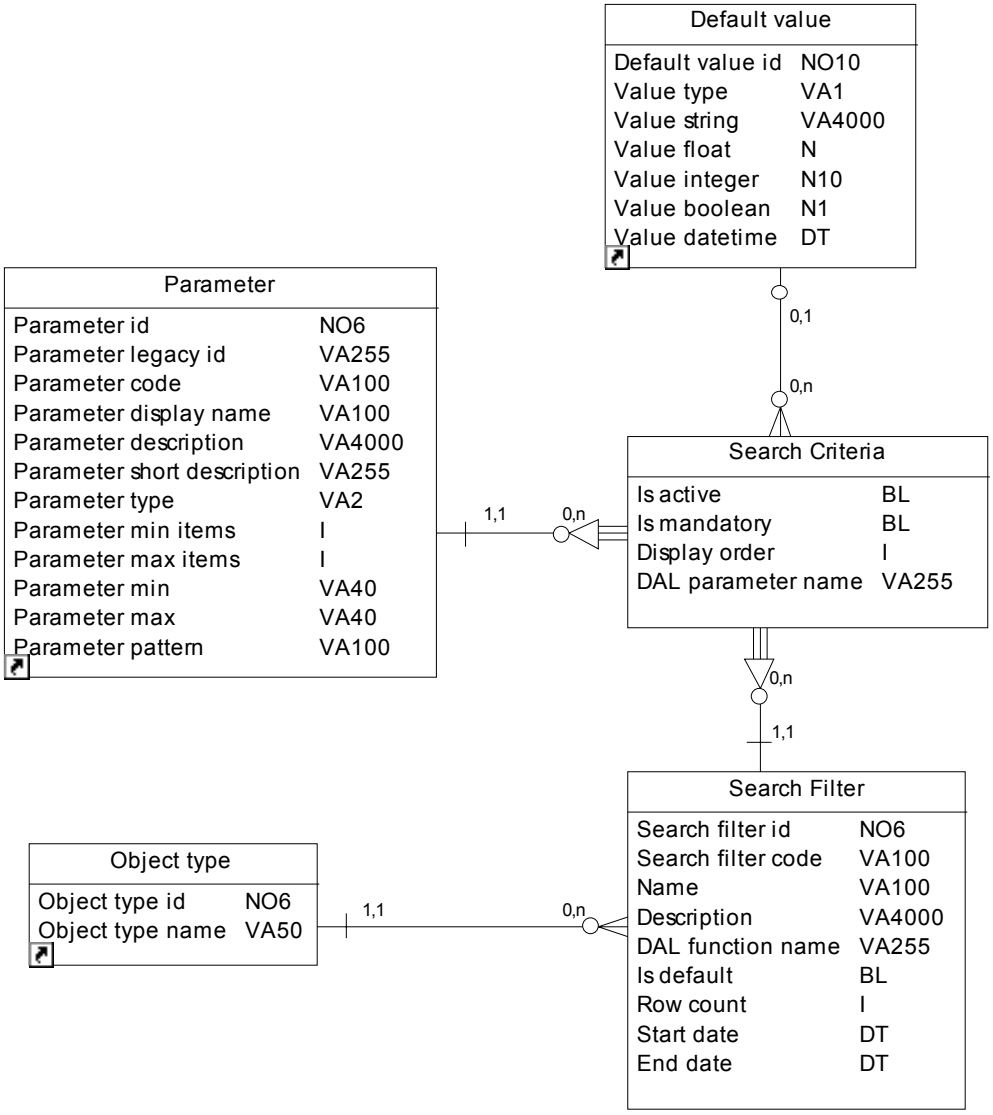
- A legacy ID
- A name
- A description
- A start date
- An end date

The minimum and maximum values for ordering a document are defined for each piece of documentation available for a given organization type:

- If MinValue is null, there is no minimum constraint (equivalent to MinValue = 1)
- If MaxValue is null, there is no maximum constraint

FILTER

Submodel Diagram



About this Submodel

This submodel defines filters. A filter is an abstraction used in the presentation layer to:

- Dynamically obtain all the criteria of a search
- Hide the complexity of the real query from the presentation layer
- Help the presentation layer to set the criteria values simply

Basic filter properties:

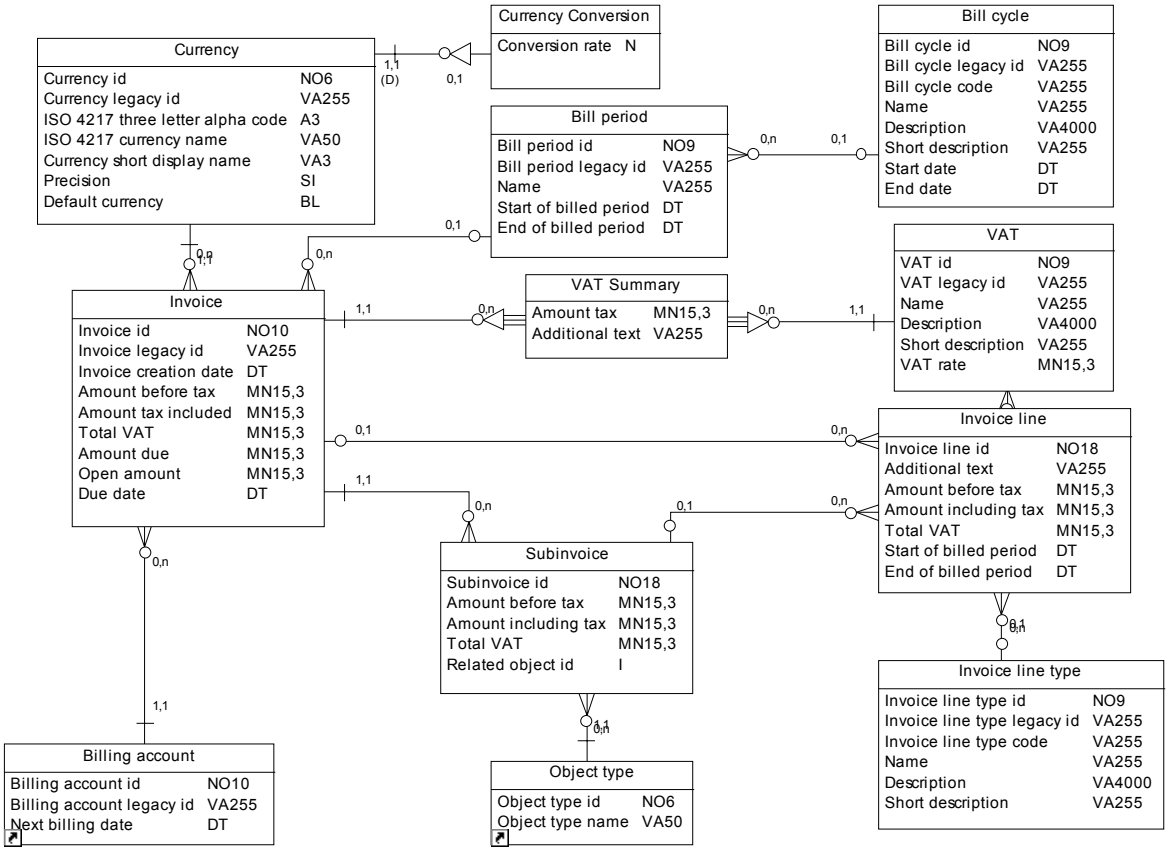
- A filter is used to search for an object of a specific type (Organization, Level, Contract…)
- A default filter can be defined for every object type.
- A filter specifies which DAL function must be used to search for the object instances.
- A filter can specify the default maximum number of records that can be selected.
- A filter specifies all search criteria.

A filter criterion is a parameter with the following characteristics:

- `IS_ACTIVE` Flag: specifies if the criteria must be used (allow customization without modifying the query)
- `IS_MANDATORY` flag: specifies if the criteria must be filled
- `DISPLAY_ORDER`: specifies the order to display criteria
- `DEFAULT_VALUE`: specifies the default value of the criteria
- `DAL_PARAMETER_NAME`: specifies the DAL parameter for mapping.

INVOICE

Submodel Diagram



About this Package

This submodel defines invoices. An invoice is the bill for usage and services over a given period of time. It has information such as invoice number, invoice date, total amount, and so on.

An invoice may contain tax lines (VATSummary).

Each invoice – VAT association is characterized by a VATSummary. This summary specifies:

- the tax amount
- an additional text (to be able to have some dynamic text along with the reference tax name). This text is not localizable

VAT is a reference object defining types of taxes.

An Invoice may contain InvoiceLines.

An InvoiceLine:

- Has some details about the invoice (for example: total subscription amount, total usage amount with invoice data lines for international calls, and so on.)
- May contain sub InvoiceLines

The InvoiceLine object has the following attributes:

- BillPeriodStart
- BillPeriodEnd (bill period covered by this line item)

An Invoice can also be analyzed on different axes (SubInvoice) like contracts, line type.

Only SubInvoice by contract are managed in the current version.

A SubInvoice, like a Invoice, may contain InvoiceLines.

An Invoice is related to a BillPeriod object (itself related to a BillCycle), provided by the billing system. A bill period of an invoice is typically the period between the last invoice and this one (bill period for the calls). This eases the display of a customer's invoices of different billing accounts that fall in the same bill period (and thus comparable between each other.)

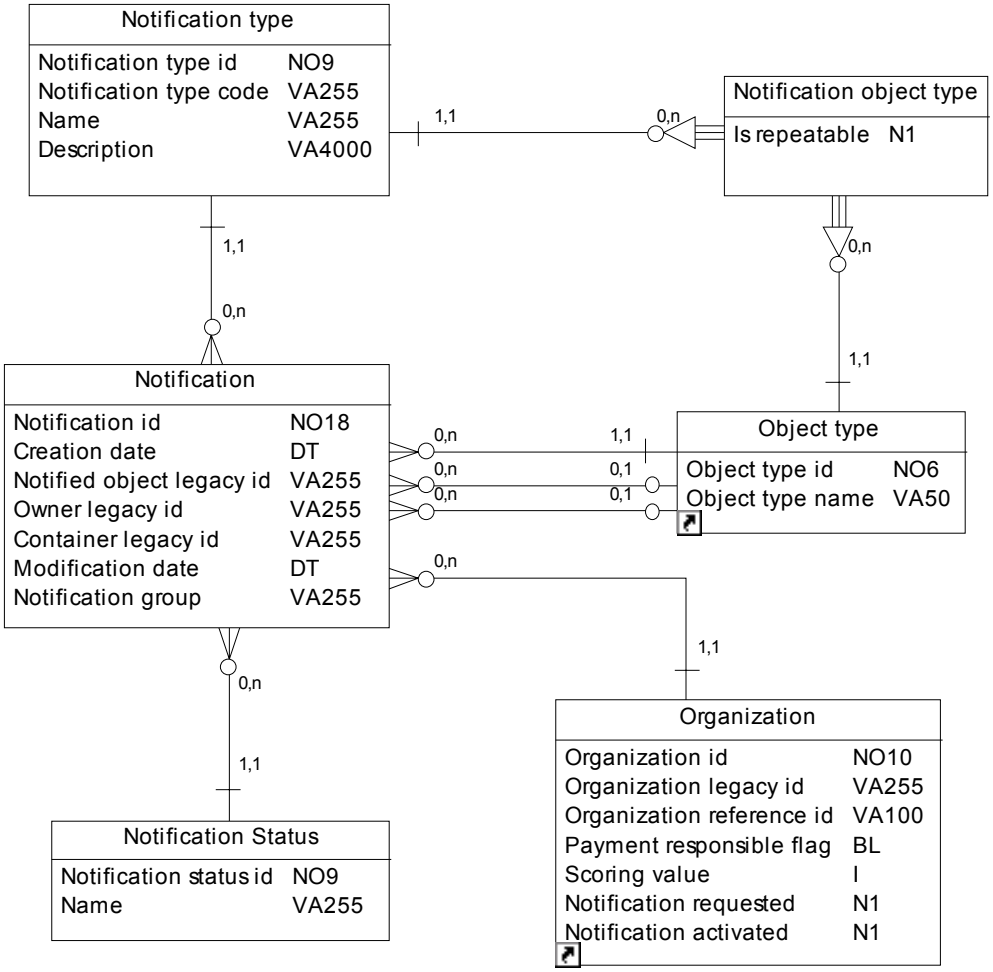
A BillCycle defines:

- A day of the month when the bill must be created
- The periodicity of the bill creation

The current bill cycle of a billing account is not managed.

NOTIFICATION

Submodel Diagram



About this Submodel

This submodel defines notifications. The notification mechanism allows the CID to notify the CBU of changes in order to synchronize data.
A notification event occurs when one of the following occurs:

- Users creating or updating an object directly in the CID
- OSS notifying the CID of object creations or updates
- Batch processes creating or updating organizations when synchronizing with the CID

The type of organization that generates a notification event when modified can be specified.
Every possible notification event type is specified as reference data with the following attributes:

- Code
- Name: name of the notification event type
- Description: specify the notification event type

An event type only specifies the type of action (create, remove, modify, update).

The `IS_REPEATABLE` flag specifies if an event must be generated if the same event having the not yet notified status for the same object is already in the notification queue.

You can configure this for every consistent couple notification type-object type. When generated, a notification event is stored in the notification queue (NOTIFICATION table) with the following attributes:

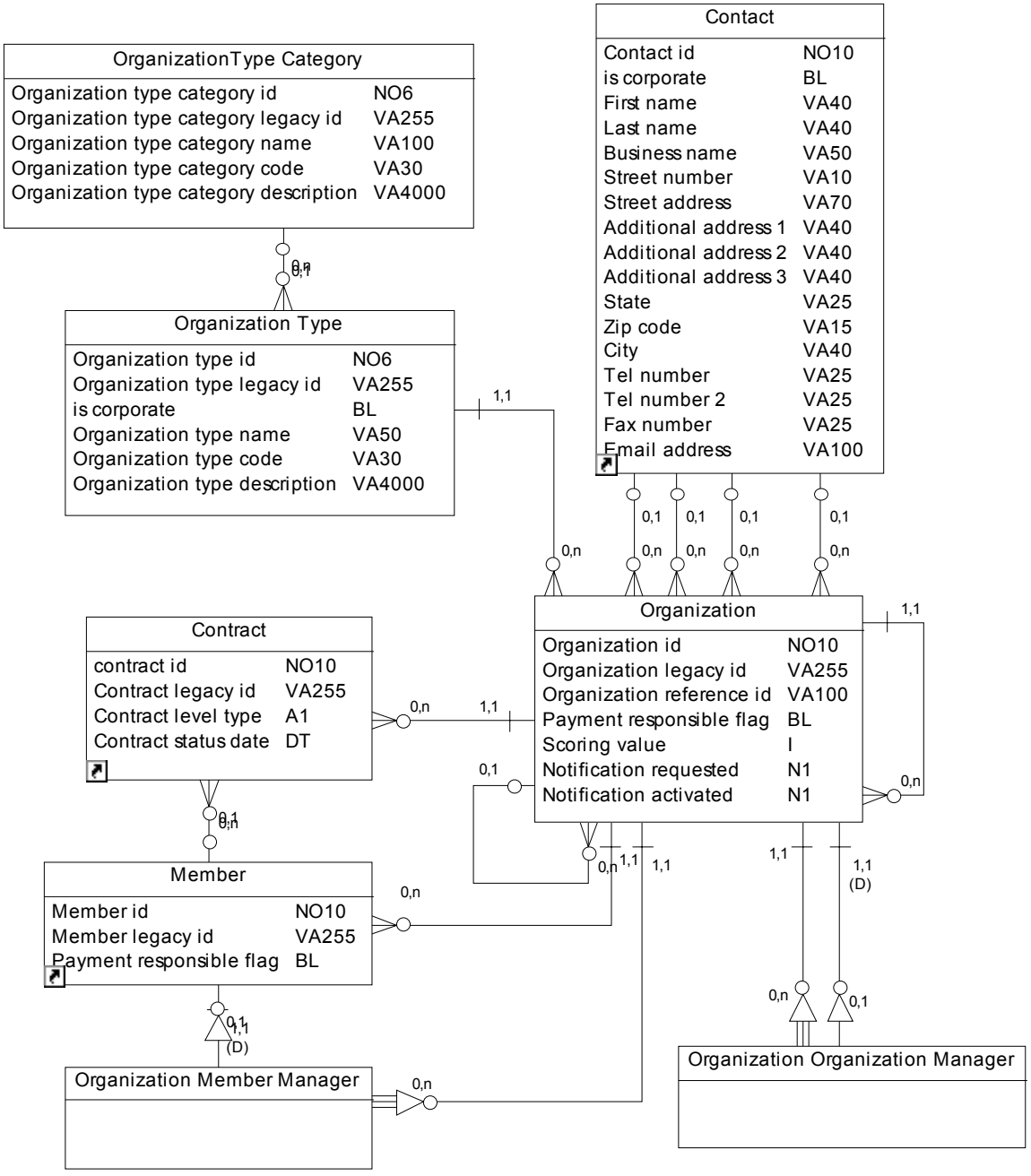
- Notification event type
- Event creation date
- Object to notify and its type
- Container legacy id and its type
- Owner legacy id and its type
- Notification group
- Status of the event in the notification queue

Possible notification statuses are:

- Not yet notified: the event has not been read by the synchronization process
- Notification in progress: the event has been read by the synchronization process but not processed
- Notification Done: event notification has been submitted
- Notification Failed: error in processing of the notification event note

ORGANIZATION

Submodel Diagram



About this Submodel

This submodel defines organizations.

An organization is an entity managed by the system which defines a customer, a telco partner, or the telco itself. The organization type differentiates the nature of the organization, for example:

- Residential customer
- Business customer
- Telco
- Dealer
- Supplier

The `IS_CORPORATE` flag (in the `ORGANIZATION_TYPE` table) indicates whether the organization is corporate or residential.

Organization types are grouped into categories. For example: residential customers and business customers can be grouped under the customer category.

An organization is defined by a legal contact and a billing contact.

An organization is described as a hierarchy. Each node of the hierarchy is a level (for example: a department or a site). Each node is defined by a legal contact, and by a billing contact.

The hierarchy terminations can be:

- Contracts
- Members

A contract can be assigned to a member of the same level (for example: the user of a mobile phone). A member can be defined by a legal contact.

An organization, a level, or a member can be responsible for payment. The `PaymentResponsible` flag (in the `PaymentResponsible` interface) indicates whether the entity is responsible for payment or not. A billing contact can be assigned to a PaymentResponsible.

Hierarchy elements (organization, levels and members) can manage other organizations.

The `NOTIFICATION_ACTIVATED` flag determines if notification events for the Customer Analytics CBU are generated for this organization.

Possible values are:

VALUE	DESCRIPTION

VALUE	DESCRIPTION
0	Notification disabled
1	Notification enabled

The Notification feature handles the value of this flag. The Notification feature uses the following algorithm when an event on the organization generates a notification:

- If the organization must be notified:
  - If the notification activated flag value is enabled, insert the notification in the notification queue
  - If the notification activated flag value is disabled:

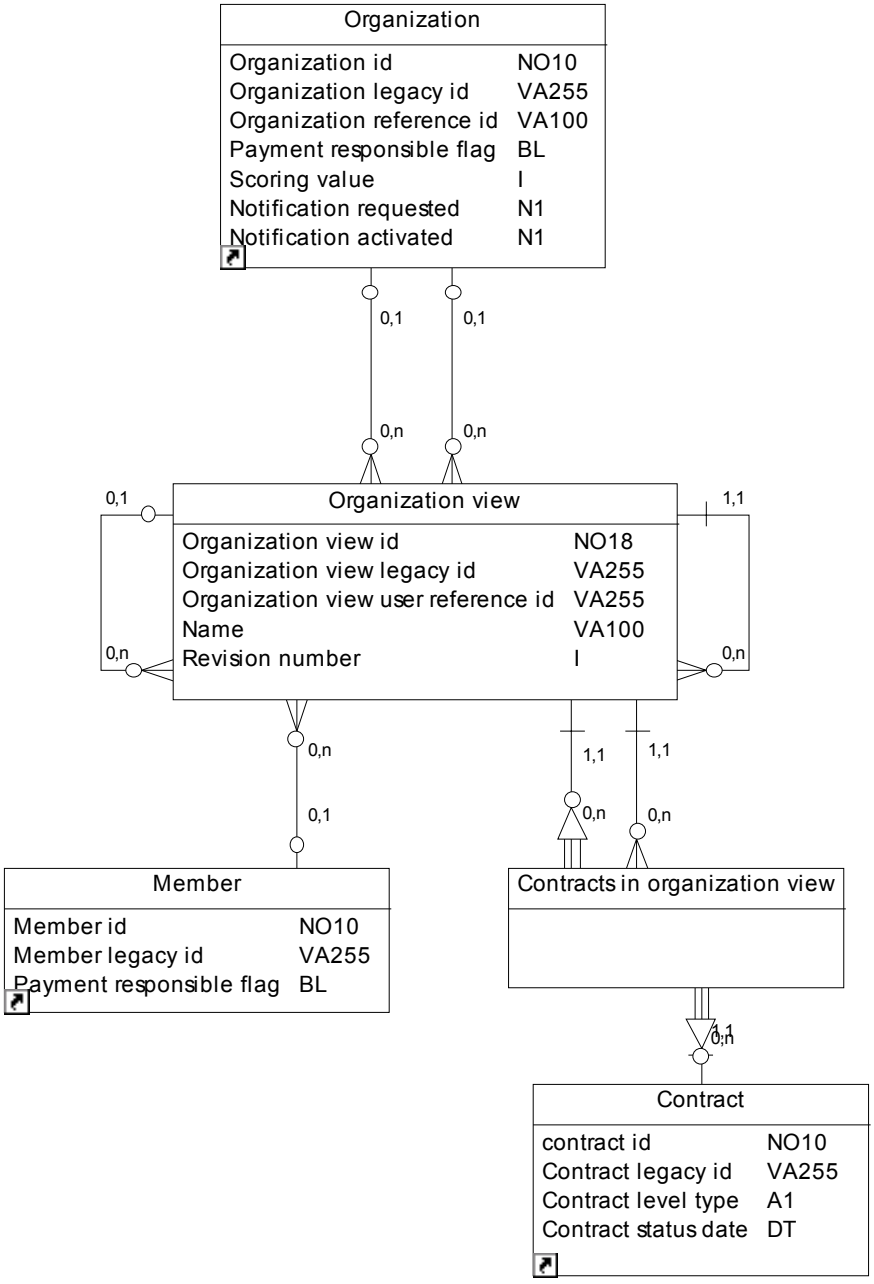
    - If the notification is a "create organization" (create level with no parent) insert in the notification queue the notification

    - Else insert an update organization notification set the flag value to notification enabled
- If the organization must not be notified:
  - If the notification activated flag value is enabled, insert a remove organization notification in the notification queue and set the flag value to disabled
  - If the notification activated flag value is disabled, do nothing

When set to 1, the NOTIFICATION REQUESTED FLAG specifies that an update organization notification must be generated for this.

ORGANIZATION VIEW

Submodel Diagram



About this Submodel

This submodel defines organization views. An organization view defines the hierarchy of a customer user contracts that is different from the known backend hierarchy. For instance, an organization can organize, display and manage contracts by location and their existing backend manages such information as departments.

A customer organization can have one or more organization view. An organization view can be associated with any level of the organization.
An organization view is a hierarchical structure with contracts associated with any node of the view (including the organization view itself).
A contract can be associated with more than one organization view.
Inside an organization view, a contract can be associated with only one organization view node for a specific time period. A start date and an end date specify the time period (the specified date included).

An organization view is specified by a user reference allowing a user to reference an organization view when uploading it. The reference must be unique for a customer organization. This reference is set by the user at organization view creation and is not modifiable.

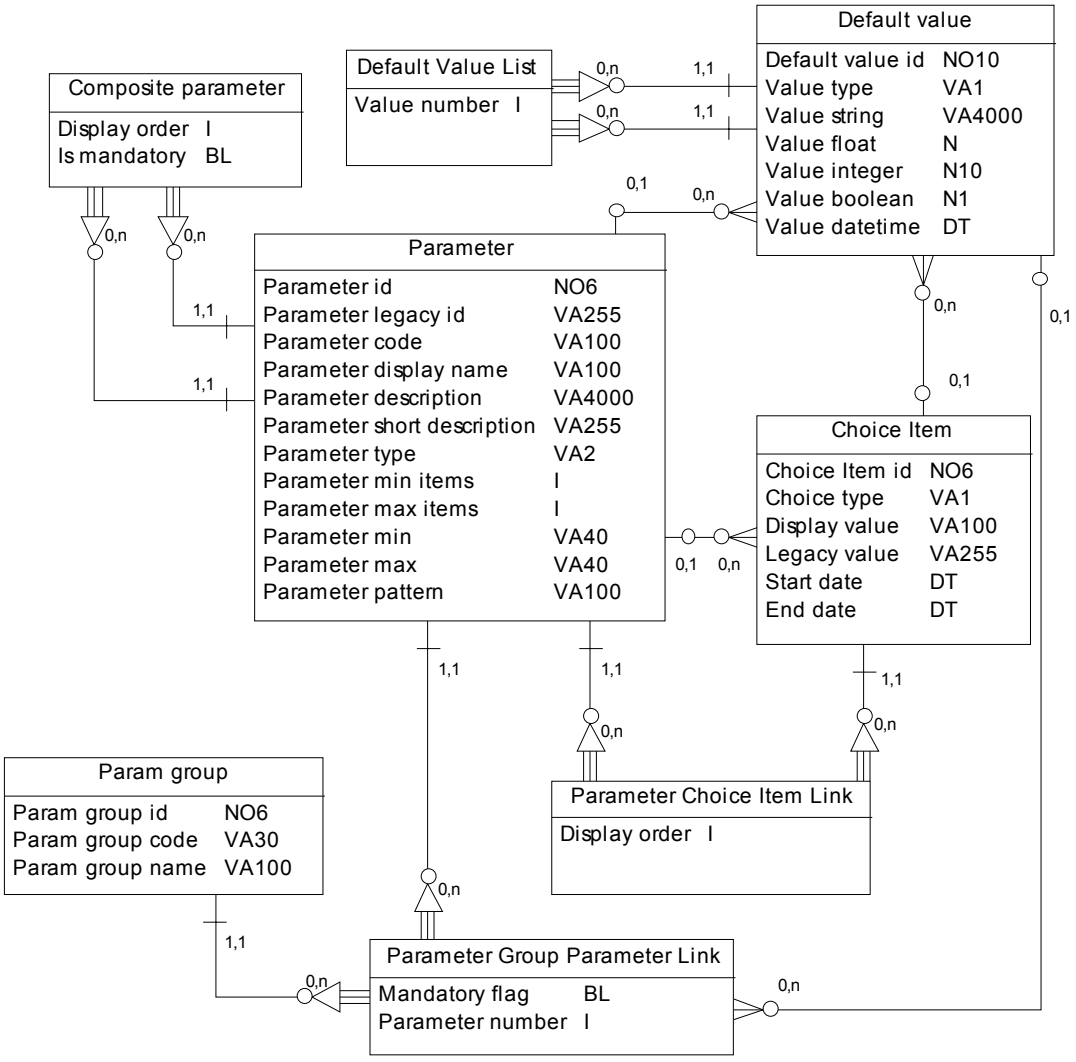
Every node of the organization view is specified by:

- Name
- Legacy id

An organization view is associated with the user who owns it (by default, the user that created the organization view).

PARAMETER

Submodel Diagram



About this Submodel

This submodel defines any parameters or additional attributes used in the system (service parameters, payment method parameters, request parameters, and so on.)

Every parameter is defined by:

- A legacy ID (known by the back end system to identify the parameter)
- A code (to manipulate the parameter)
- A name (to display the parameter)
- A long and a short description (to display the parameter)
- Min and a max items numbers (these numbers define the min and max values of the parameter - list size: if max=1 the parameter is not a list)
- A parameter type

A parameter can be elementary or composite. An elementary parameter is specified by the following attributes:

- Pattern (only used for string parameters)
- Min and max values

Parameter Types

The following table presents the different supported parameter types:

CODE	DESCRIPTION
c	Choice
s	String
d	Decimal
l	Integer
B	Boolean
D	Date
T	Time
DT	Timestamp (date and time)
CO	Composite
DY	Dynamic

Choice Parameters

The list of allowed choices can be:

- Choice constant items (display value and legacy value)
- Choice parameter items (parameter)

Simple Parameters

To declare a simple parameter

Insert a record in the PARAMETER table and specify the type of parameter.

Example: creating a simple integer parameter with an allowed min value equal to 10:

ID	LEGACY ID	CODE	TYPE	NAME	MIN_ITEMS	MAX_ITEMS	MIN	MAX
1000	Leg_1000	Param1000	I	Param name		1	10	

To declare a simple list parameter

In the PARAMETER table, add a record having the type you need and specify the max number of items on the list.

Example: creating a list of integers with a maximum number of 10 items

ID	LEGACY ID	CODE	TYPE	NAME	MIN_ITEMS	MAX_ITEMS	MIN	MAX
1001	Leg_1001	Param1001	I	Param name		10		

To declare a choice parameter

1 Insert a C type record in the PARAMETER table.

In the PARAMETER table, the Max column is the maximum number of choice items that can be selected. To create a mono choice parameter (a parameter that can have only one value among a set of possible values), you set the max value to 1.

Example: declaring a multi choice parameter with a max of 5 selectable choice items:

ID	LEGACY ID	CODE	TYPE	NAME	MIN_ITEMS	MAX_ITEMS	MIN	MAX
1003	Leg_1003	Param1003	C	Param name		1		5

Example of a mono choice parameter: creating a list of integers with a maximum number of 10 items

ID	LEGACY ID	CODE	TYPE	NAME	MIN_ITEMS	MAX_ITEMS	MIN	MAX
1002	Leg_1002	Param1002	C	Param name		1		1

2 Insert a record in the `CHOICE_ITEM` table for each choice. If the choice is already in the table, you do not have to create it again.

Example: storing a constant choice item valid until January 2010

CHOICE ITEM ID	TYPE	DISPLAY VALUE	LEGACY VALUE	CHOICE PARAM ID	START DATE	END DATE
1000	C	Display1000	Leg1000			01/01/2010

Example: storing a parameter choice item (the choice is itself a parameter having an id of 1100)

CHOICE ITEM ID	TYPE	DISPLAY VALUE	LEGACY VALUE	CHOICE PARAM ID	START DATE	END DATE
1001	P		Leg1001	1100		

3 Associate the choice items and parameters by entering a record in the `PARAMETER_CHOICE_ITEM_LINK` table.

Example: creating a list of integers with a maximum number of 10 items

PARAM ID	CHOICE ITEM ID	DISPLAY ORDER
1002	1000	0
1002	1001	1

To declare a composite parameter

1 Insert a CO type record in the `PARAMETER` table.

To declare a composite list parameter, set the `MAX_ITEMS` value to the number of possible choices.

ID	LEGACY ID	CODE	TYPE	NAME	MIN_ITEMS	MAX_ITEMS	MIN	MAX
1004	Leg_1004	Param1004	CO	Param name		1		

2 Declare the parameters that are part of the composite parameter.

Example: declare a simple string and simple integer parameter that will be part of the composite parameter.

ID	LEGACY ID	CODE	TYPE	NAME	MIN_ITEMS	MAX_ITEMS	MIN	MAX
1005	Leg_1005	Param1005	S	Param nameS		1		
1006	Leg_1006	Param1006	I	Param nameI		1		

3 Insert records in the `COMPOSITE_PARAMETER` to associate the composite parameter with its parameters.

Example: linking the string and integer parameters to the composite one and declaring the Param1005 as mandatory

PARAM ID	SUB PARAM ID	MANDATORY FLAG	DISPLAY ORDER
Param1004	1005	1	0
Param1004	1006	0	1

Parameter Values

The following table presents the different storage value formats:

PARAMETER VALUE TYPES	
S	Simple (used to store simple values such as int, float, string,...)
L	List
H	Choice
O	Composite

To store a simple value

Create a record of type S in the default value table.

DEFAULT VALUE

Example: Storing an integer value:

DEFAULT VALUE ID	CHOICE ITEM ID	PARAM ID	VALUE TYPE	VALUE STRING	VALUE FLOAT	VALUE INTEGER	VALUE BOOLEAN	VALUE DATE TIME
1			S			Value		

To store a list of values

1 Create a record of type L in the default value table

2 Create one record per value in the list in the default value table

3 Associate the first record and the subsequent records via the default value list table

4 If required, enter the order of the retrieved values in the value number column

DEFAULT VALUE

Example: Storing a list of string values:

DEFAULT VALUE ID	CHOICE ITEM ID	PARAM ID	VALUE TYPE	VALUE STRING	VALUE FLOAT	VALUE INTEGER	VALUE BOOLEAN	VALUE DATE TIME
1			L					
2			S	Value1				
3			S	Value2				

DEFAULT VALUE LIST

DEFAULT VALUE ID	DEFAULT SUB VALUE ID	VALUE NUMBER
1	2	
1	3	

To store a choice value

1 Create a record of type H in the default value table

2 Create one record per selected value in the choice:
If the selected value is itself a parameter:

- Put in the Value type a type consistent with the choice item parameter type (L, H or O)
- Create all values of the choice item parameter (as explained in this chapter) depending on the choice item parameter type.

3 Associate the first record and the subsequent records via the default value list

4 If required, enter the order of the retrieved values in the value number column

DEFAULT VALUE

Example: Storing a choice with a choice item which is itself a choice parameter:

DEFAULT VALUE ID	CHOICE ITEM ID	PARAM ID	VALUE TYPE	VALUE STRING	VALUE FLOAT	VALUE INTEGER	VALUE BOOLEAN	VALUE DATE TIME
1			H					
2	Item1_Id		S					
3	Item2_Id		H					
4	Sub Item1 Id		S					
5	Sub Item2 Id		S					

DEFAULT VALUE LIST

DEFAULT VALUE ID	DEFAULT SUB VALUE ID	VALUE NUMBER
1	2	
1	3	
3	4	
3	5	

To store a composite value

1. Create a record of type O in the default value table
2. In the default value table create one record per parameter in the composite value and relate each to its parameter
3. Associate the first record and the subsequent records via the default value list table
4. If required, enter the order of the retrieved values in the value number column

DEFAULT VALUE

Example: Storing a composite parameter value composed of two string parameters:

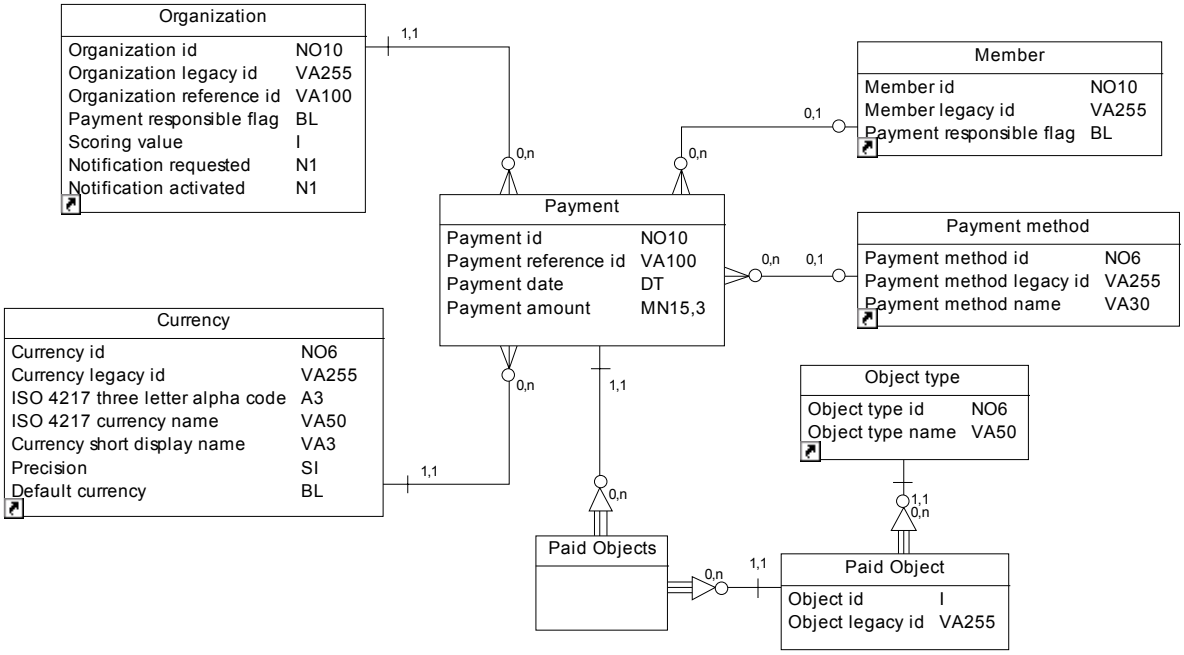
DEFAULT VALUE ID	CHOICE ITEM ID	PARAM ID	VALUE TYPE	VALUE STRING	VALUE FLOAT	VALUE INTEGER	VALUE BOOLEAN	VALUE DATE TIME
1			O					
2		Par1_Id	S	Value1				
3		Par2_Id	S	Value2				

DEFAULT VALUE LIST

DEFAULT VALUE ID	DEFAULT SUB VALUE ID	VALUE NUMBER
1	2	
1	3	

PAYMENT

Submodel Diagram



About this Submodel

This submodel defines payments. Payments are defined by:

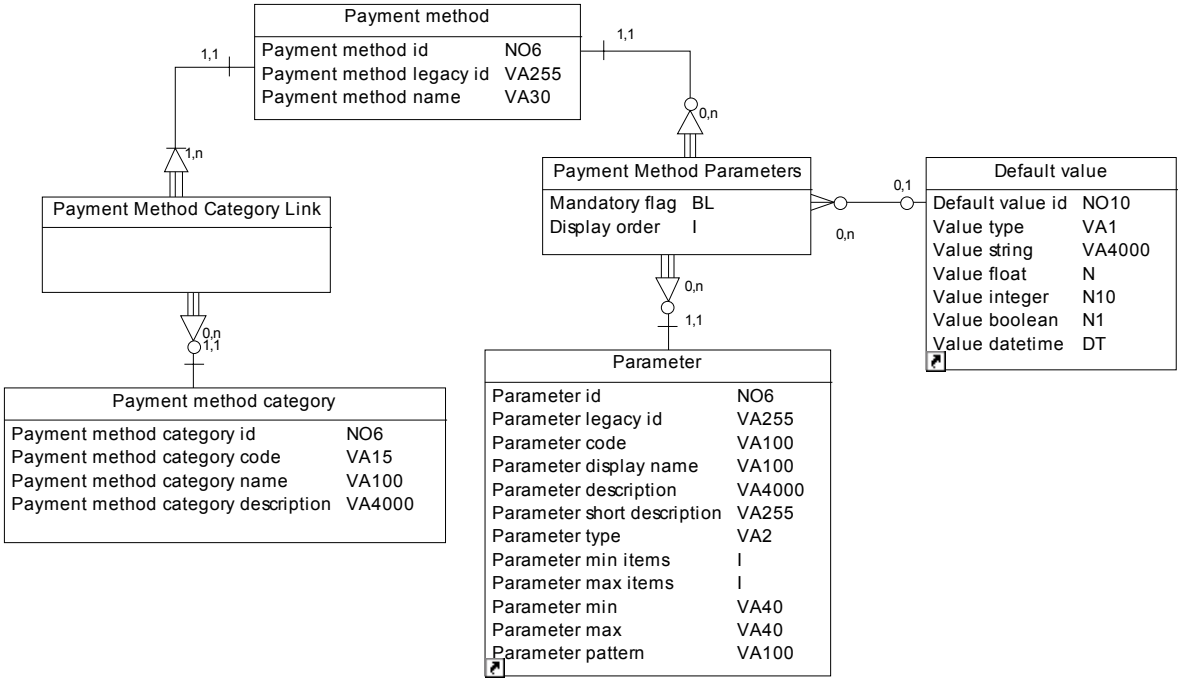
- Payment reference
- Payment method
- Payment value
- Payment currency
- Payment date

Each payment is related to the objects for which it has paid (for the current version: invoices or prepaid packages).

A payment can be related to more than one object, for example: one payment for two invoices. An object can be paid for with more than one payment, for example: a bill can be paid in two installments.

PAYMENT METHOD

Submodel Diagram



About this Submodel

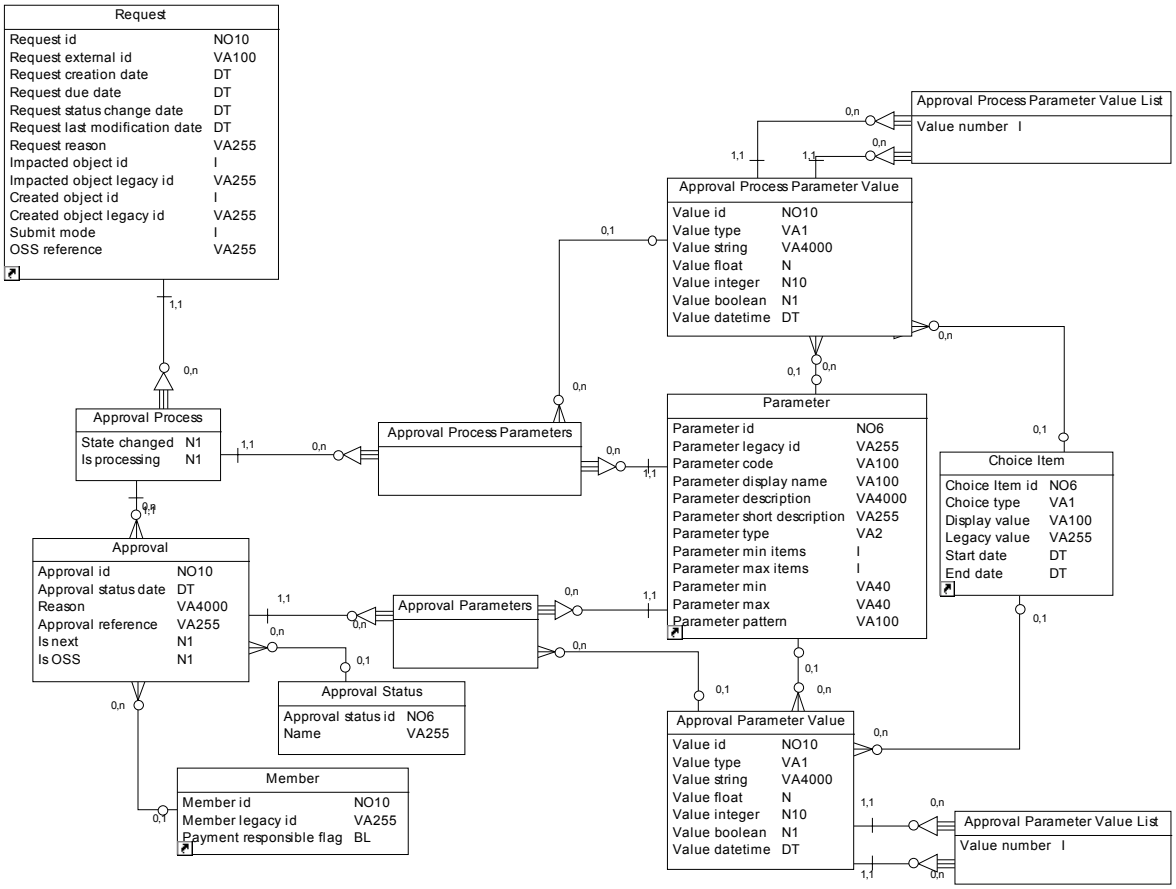
This package defines payment methods. A payment method is defined by the list of parameters required when using it (in Payment Method Parameters). For example, if the payment method is credit card, the payment method parameters are *credit card company*, *credit card number* and *credit card validity date*)

Payment methods are grouped into categories to allow them to be selected according to need. For example, when using the recharge prepaid card feature, the list of available payment methods can be filtered by using the recharge prepaid payment method category.

A payment method can belong to more than one category.

PROCUREMENT

Submodel Diagram



About this Submodel

This submodel defines procurement items such as approval processes. An approval process is the process of approving a request.

This process is defined by:

- Any parameters required to qualify and execute the approval process
- An ordinate list of approvals defining every step of the approval process
- A status indicating whether there has been an evolution of the approval process since the last evaluation (State Changed)
- A status indicating whether the approval process is currently being evaluated (Is processing)

Every approval is defined by:

- An approver

  The approver can be:

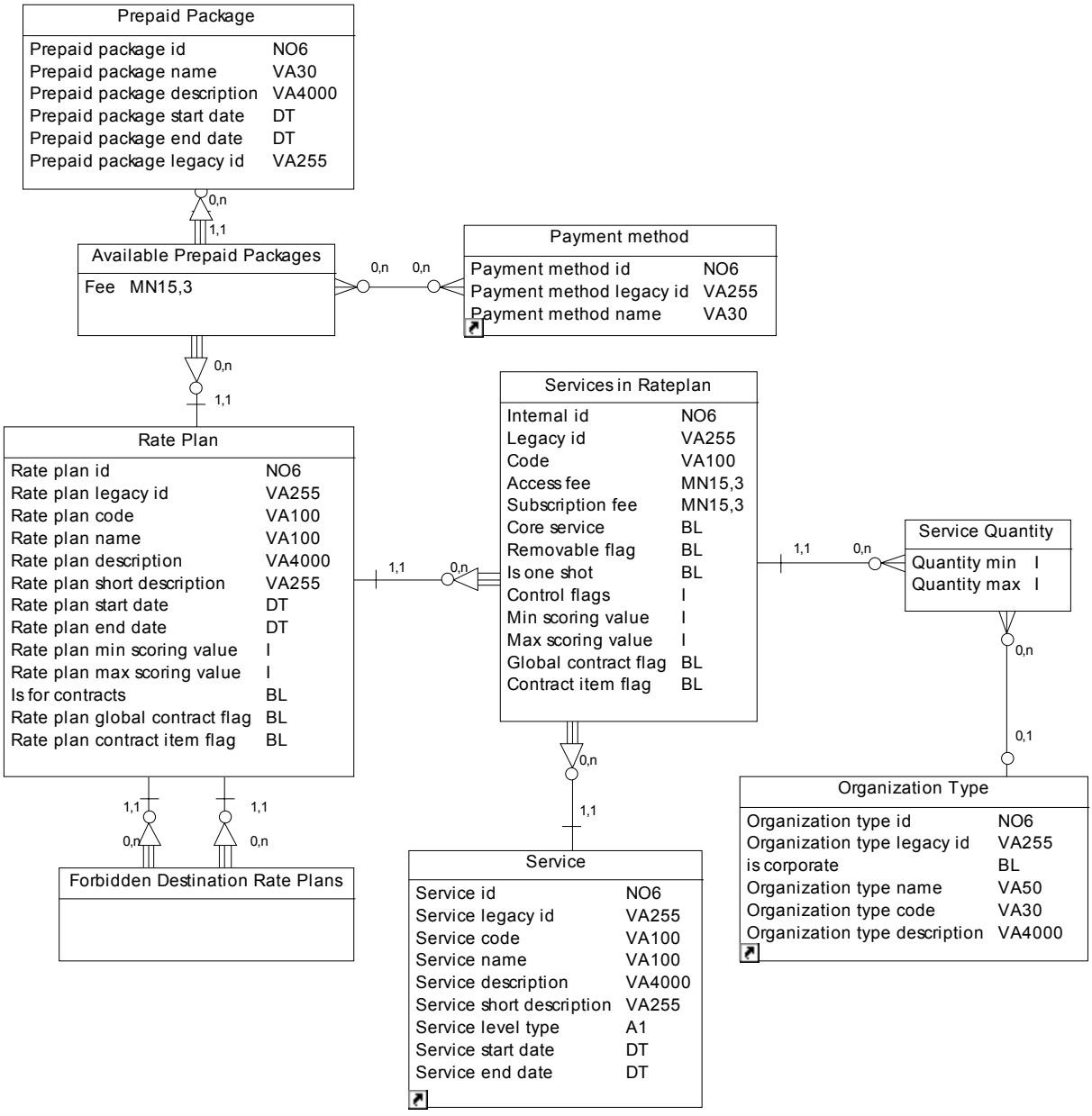
  - a member:
  - an OSS (when `Is OSS` is set to true and OSS reference is set in Approval reference)
  - an anonymous user (when `Is OSS` is set to false and user reference is set in Approval reference)

- A status that can be:
  - To be approved
  - Approved
  - Denied

- A status date
- A reason
- Any specific parameters of the approval
- A flag to specify whether this approval must be confirmed to resume the evaluation process (Is Next)

RATEPLAN

Submodel Diagram



About this Submodel

This submodel defines rate plans. A rate plan defines the sale policy of a subset of the offer's services.

A rate plan can be defined in more than one offer. For example, a dedicated offer can have rate plans in common with a public offer, in addition to certain specific rate plans.

Validity of a rate plan in an offer depends on the start and end dates of its association with the offer.

Services available in a rate plan are defined in `SERVICES IN RATEPLAN`. It defines the fees of the service and whether it is core or mandatory.

The rate plan fee is equal to the sum of the fees of the mandatory services.

The `SERVICES IN RATEPLAN` object contains a `REMOVABLE` flag. In the current version, this flag is information only for use by JSPs. There is no business logic relating to it in the BLM.

For a specific contract, the availability of a rate plan depends on:

- The contract level (global or contract item)
- The scoring value of the customer
- The current rate plan

`Forbidden Destination Rate Plans` defines rate plans which cannot be subscribed in conjunction with the current rate plan.

There are two categories of rate plan:

- Rate plans specifying the contract sale policy (for example rate plan 1 hour + 1 hour)
- Rate plans defining sale policy independently of the contract (for example: handset cost)

The category to which a rate plan belongs is defined by the 'is for contract' rate plan attribute.

For a specific contract with a specific rate plan, the availability of a service can depend on the scoring value of the customer, and on the contract level.

A service within a rate plan can include a number of free units. Free units are defined by:

- Number of units
- A package type
- Number of free units during the day
- Number of free units during the night
- Number of free units at the weekend
- etc.

A unit can be:

- Minutes
- Money
- Bytes
- etc.

The list of prepaid packages (for the ‘recharge prepaid contract’ feature) depends on the payment method and on the rate plan. Equally, the list of available payment methods depends on the prepaid packages.

A service can be sold as a "one-shot" service or as a recurrent one. The `IS ONE SHOT` flag specifies the chosen policy for the rate plan. The quantity of one-shot services ordered must fall within the min quantity and max quantity declared in the rate plan sale policy. These quantities can depend on the recipient organization type.

Rate Plan Service has a `CONTROL FLAGS` column for specific controls. `CONTROL FLAGS` is used to specify whether the same controls apply for a one-shot service as for a recurrent service. For example, The user cannot change a service if there is a pending change request.

The first 8 bits of `CONTROL FLAGS` are reserved.

Every rate plan service can be referenced by a legacy ID (if required by the back office system).

A service can include a number of free units within a rate plan.

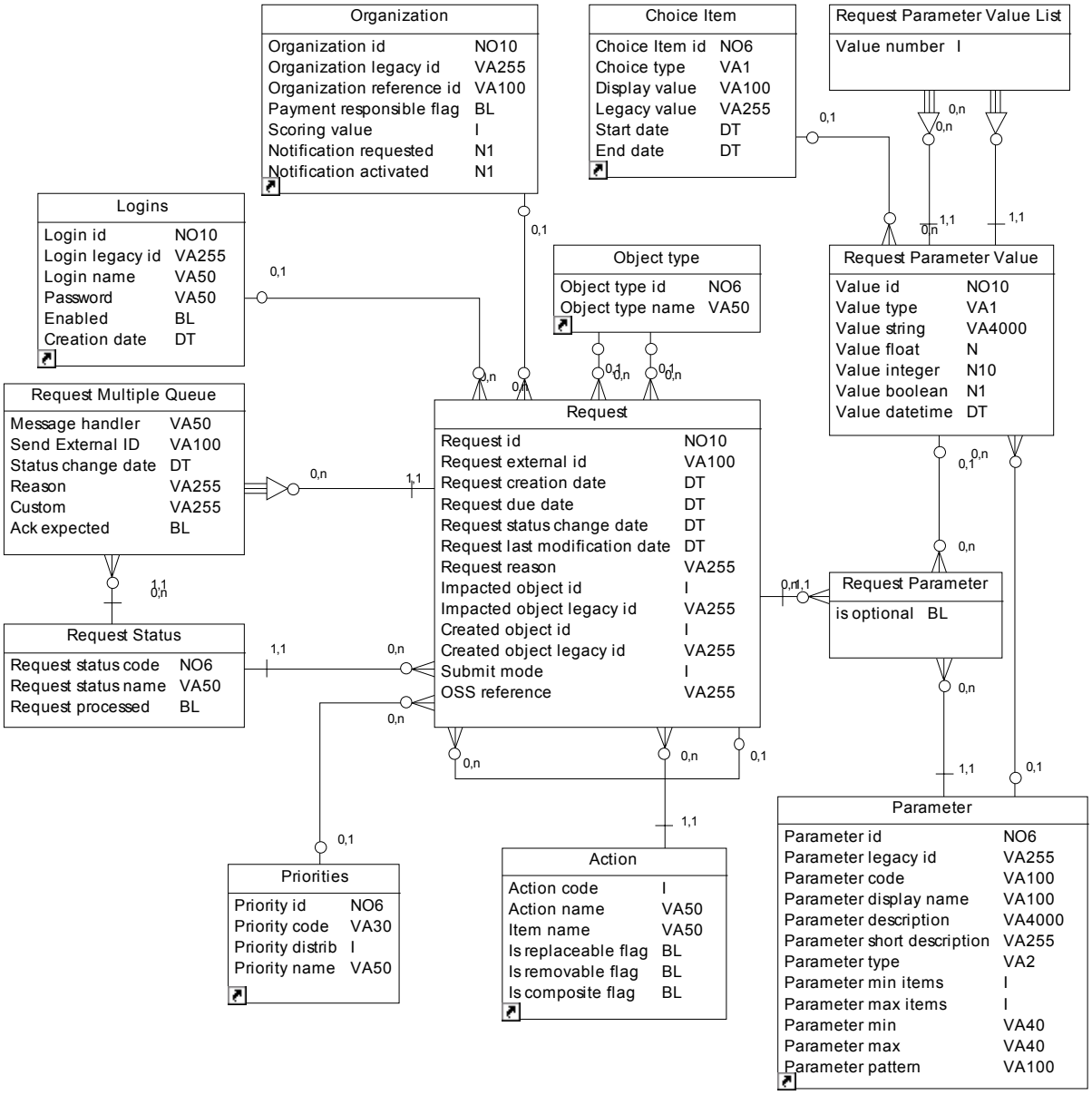
These free units are defined by:

- Number of units
- A package type
- A unit

The list of prepaid packages depends on the payment methods. Similarly, the list of available payment methods depends on the prepaid packages.

REQUEST

Submodel Diagram



About this Submodel

This submodel defines requests. When a user makes a change that impacts one of the legacy systems (such as add or remove service), the request is stored in the `REQUEST` table. The parameters of the request are stored in the `REQUEST_PARAM` table. A request can be composite or elementary.

The following list describes requests:

- A request can be elementary or composite
- A composite request is a set of requests that you must submit together (for consistency)
- A request is linked to the user who generated it
- A request is related to the impacted object (hierarchy node, billing account, contract or member)
- With certain requests (for example: add contract requests), the Synchronizer connector stores the ID and type of the created object in the request
- A request can be submitted in synchronous or asynchronous mode
- The Synchronizer connector can also store a functional OSS reference received in an acknowledge message for JSP display purposes

REQUEST_STATUS

The following table presents request statuses:

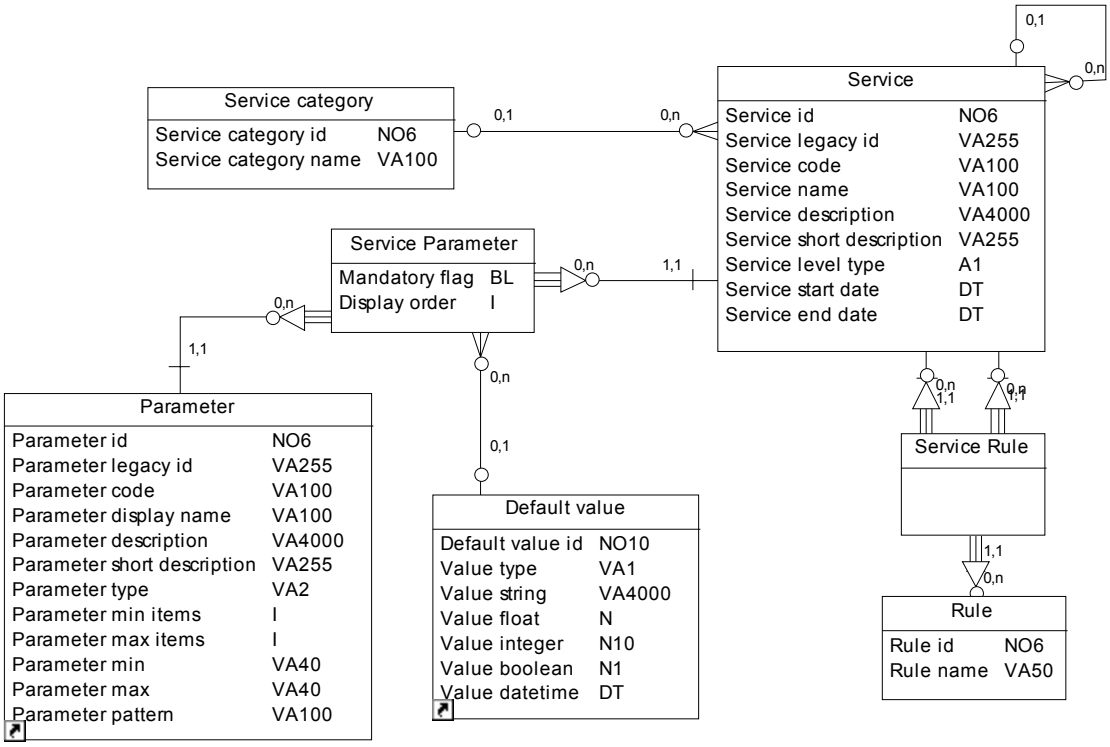
REQUEST STATUS CODE	REQUEST STATUS NAME
1	To be approved
2	Not yet submitted
3	Submitted
4	Done
5	Failed
7	Denied
8	Submission in progress
10	Transport Failed
12	Acknowledged

Pending requests are requests with one of the following statuses:

- Not Yet Submitted
- Submission in progress
- Acknowledged
- Transport failed
- Submitted
- To be approved

SERVICE

Submodel Diagram



About this Submodel

This submodel defines services. A service defines the core and non-compulsory options available for a contract (for example: Telephony, Call forwarding…). Services may be shared between different contract types.

A service category is a consistent set of services used for display purposes (for example: GSM services, value added services...).

The Service Rule entity defines compatibility rules between services.

Two compatibility rules are handled:

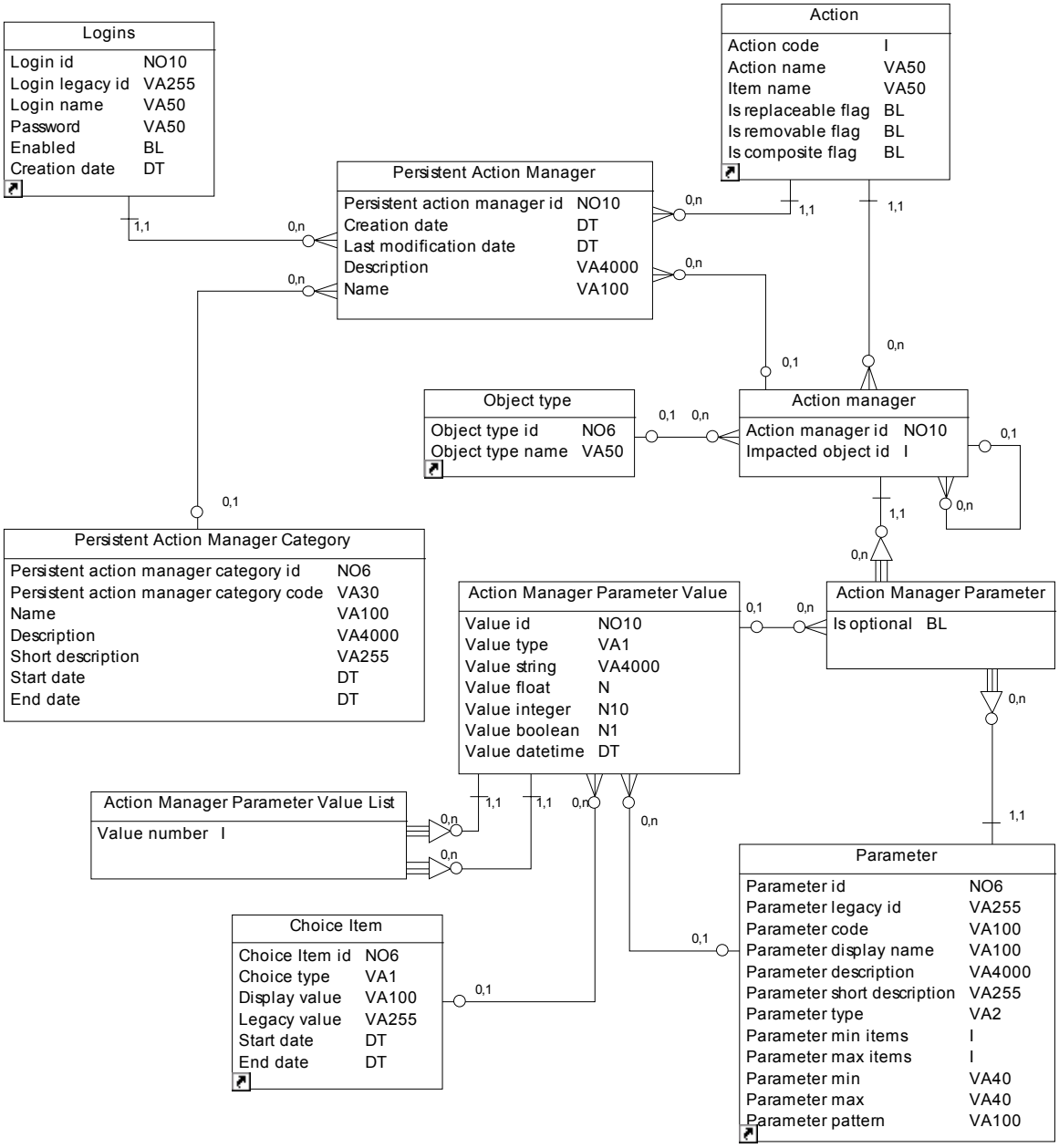
- Service 1 needs service 2
- Service 1 is incompatible with service 2

The Rule entity defines these two rules.

RULE_ID	RULE_NAME
1	Needs
2	Is incompatible with

SHOPPING CART

Submodel Diagram



About this Submodel

This submodel defines shopping carts. A shopping cart is a feature that allows a user to save items temporarily so they can be then be ordered together.

This submodel contains all the tables used to store the objects related to the persistence of a shopping cart:

- **Persistent Action Manager**

  A persistent Action manager is the persistent representation of an action manager. It is the technical implementation of a persistent shopping cart
- **Persistent Action Manager Category**

  A Persistent Action Manager Category can be used to specify the future use of a Persisted Action Manager (back up, contract template...)
- **Action manager related to the Persistent action manager**
- **Link with other objects (action and user)**
  - **Action:** root action of the persisted action manager (order, create organization, ...)
  - **User:** User who created the persistent shopping cart

A shopping cart can be saved during any processes using the shopping cart.

A shopping cart can be saved for one of the following reasons:

- To back up the current shopping cart in case of a lost session
- To be used as template for creating a new shopping cart

There is only one persistent copy of a specific shopping cart (no history).

A shopping cart can be saved many times during an order process (for example: each time the order is modified).

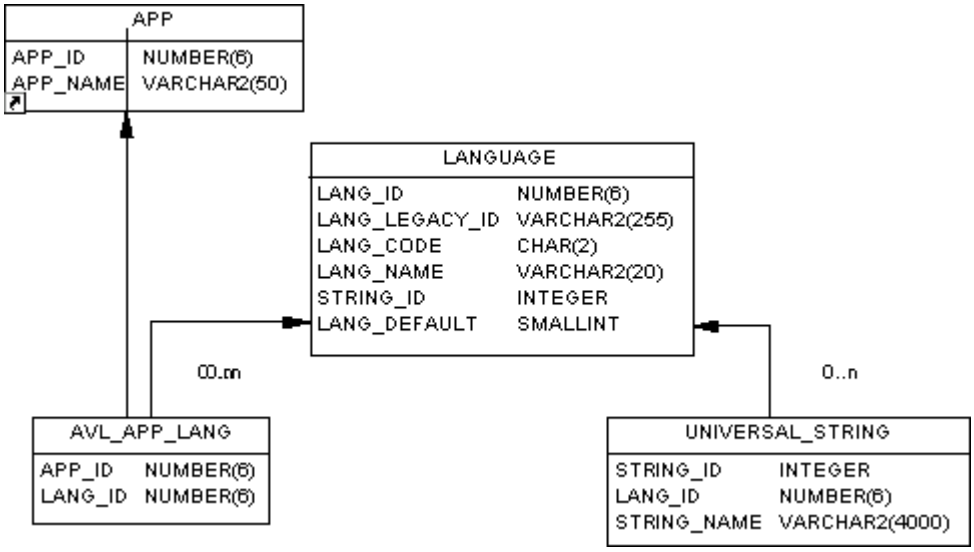
By default, there is no limit on the number of persistent shopping carts per user.

Depending on the future use of the persistent shopping cart, the user can specify:

- Name (to identify the persistent shopping cart)
- Description (to describe it)
- Link with persistent action manager category (to specify the future use)
- Any additional information to be used as criteria to retrieve the shopping cart

STRING LOCALIZATION

Submodel Diagram



About this Submodel

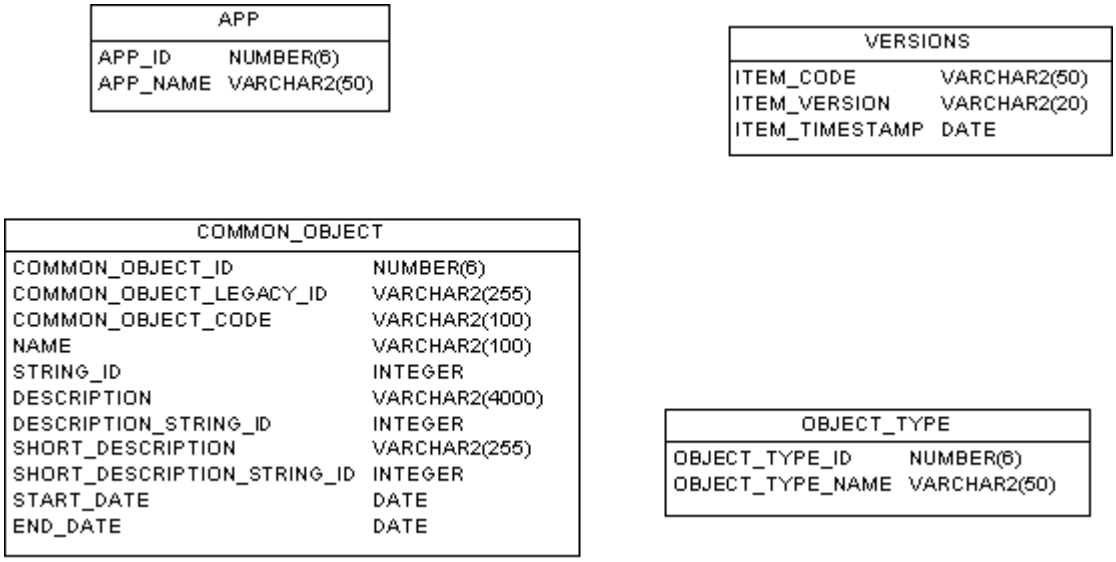
This submodel defines string localization. A string is identified by a string ID, and its localizations in different languages are stored in the UNIVERSAL_STRING table.

Each name or description column has a corresponding string ID in the same table.

If this string ID is populated, the localized name is retrieved. If not, the name or description itself is retrieved.

SYSTEM

Submodel Diagram



About this Submodel

This submodel defines system information.

The `VERSIONS` table contains information about the version of the system and application reference data.

This table contains information on:

- The item code
- The item version
- The system timestamp

The `APP` table contains information about applications.

The `OBJECT_TYPE` table contains information on the different system object types.

`COMMON_OBJECT` table contains information on common objects. A common object is used to specify any reference object with no specific business logic. For instance, it is used to specify the OSS approvers for order validation.

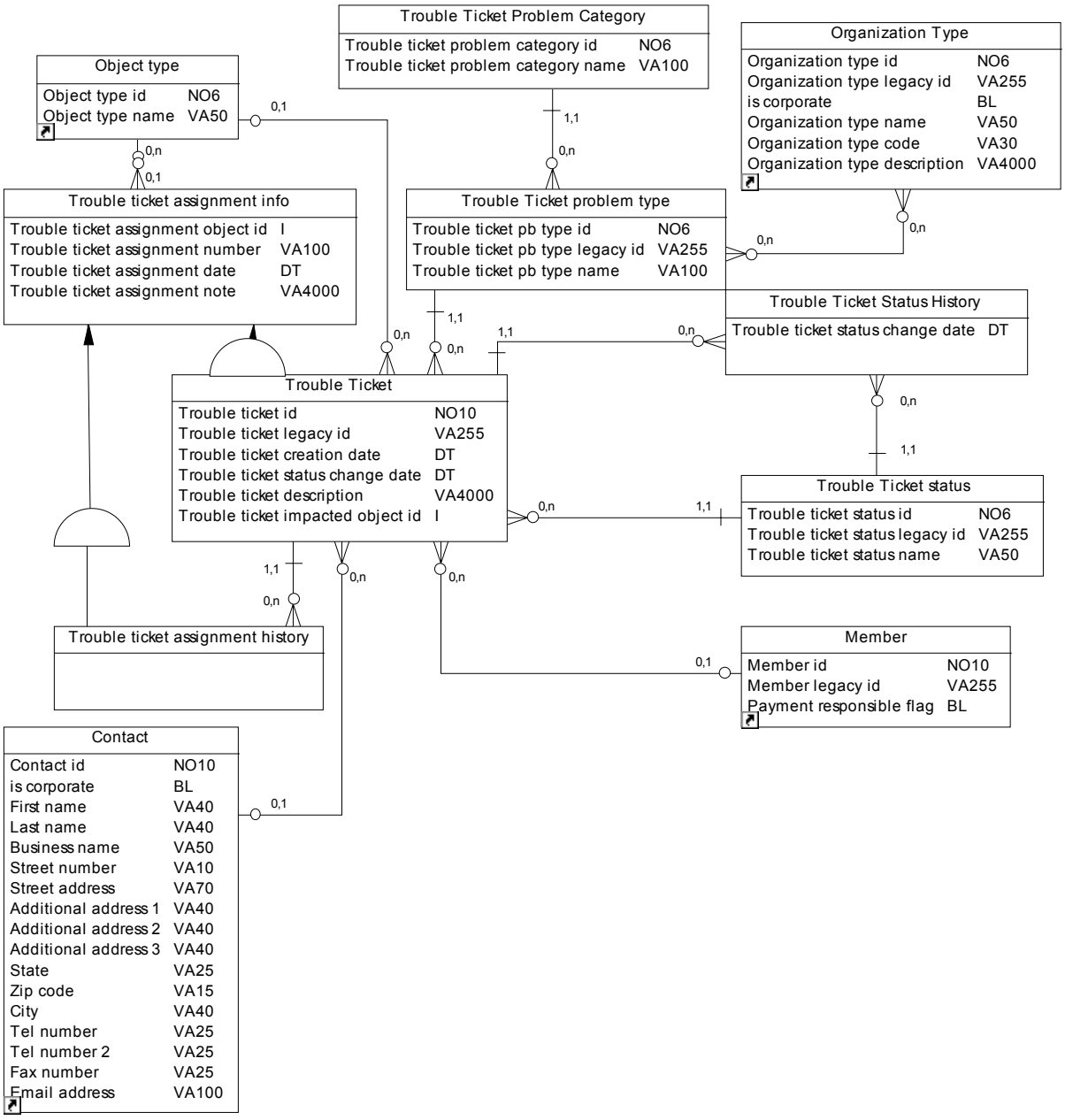
OBJECT_TYPE Table

The OBJECT_ TYPE table defines the main object types which can be used with an object ID to reference an instance of an object.

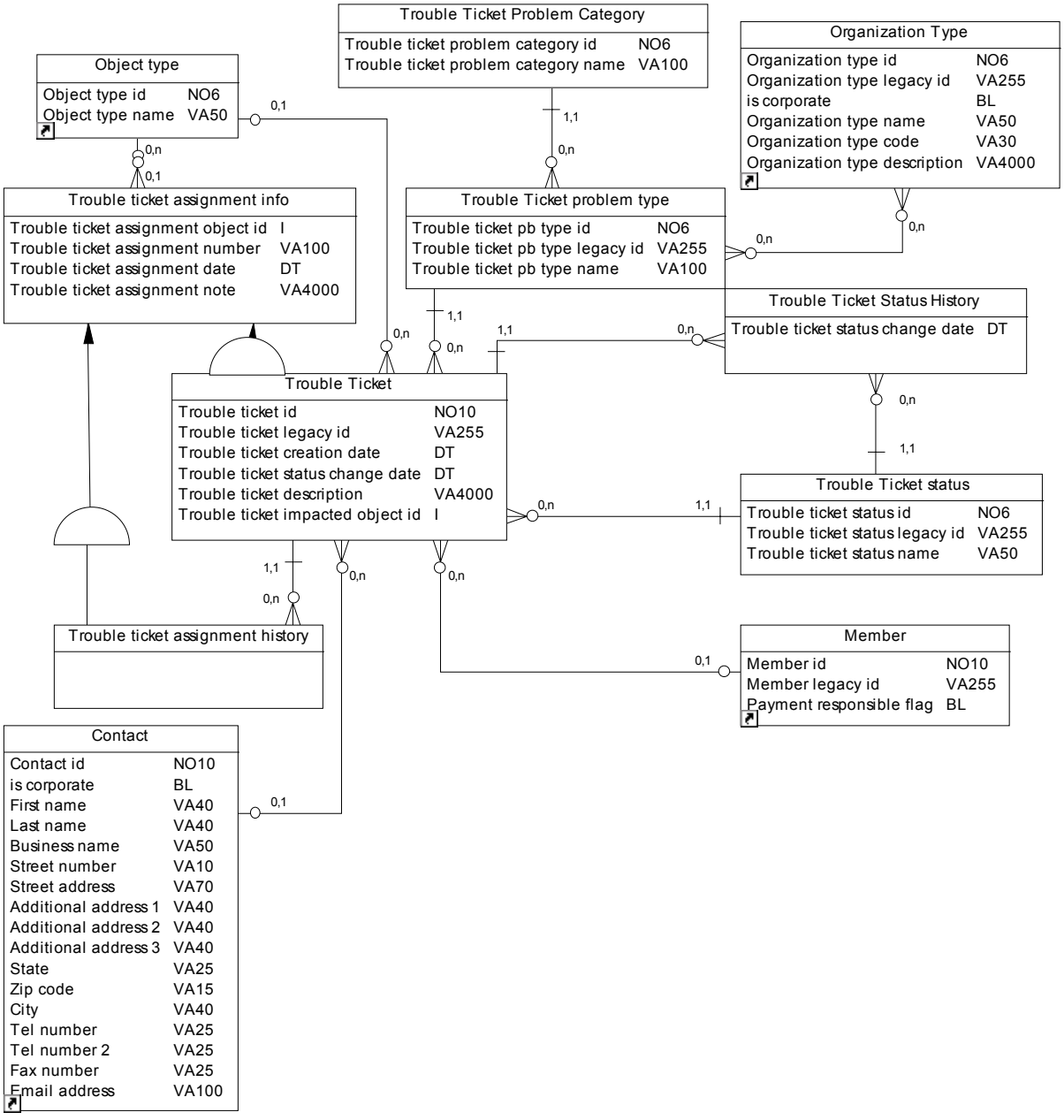
For example: if the OBJECT_ TYPE_ ID = 4 and the OBJECT_ ID = 1003, this references the contract defined by the ID 1003.

OBJECT TYPE ID	OBJECT TYPE NAME
1	Level
3	Member
4	Contract
5	Contact
6	Invoice
7	Prepaid package
8	Billing account
9	Trouble ticket
10	User
11	Persistent action manager
12	User event
13	Service
14	Request
15	Bill period
16	Contract subinvoice
17	Organization view
18	Organization view level

TROUBLE TICKET



Submodel Diagram



About this Submodel

This submodel defines trouble tickets.

A trouble ticket is defined by:

- A legacy ID
- A creation date
- A problem type
- A description of the problem
- The organization contact who created the trouble ticket
- The object impacted by the problem (hierarchy node, member, contract or invoice)
- The current trouble ticket status and status history
- The member who generated the trouble ticket (if known in the CID)

A trouble ticket can be assigned to:

- A hierarchy node (for example: a supplier company)
- A specific member (for example: a telco technician known in the CID)
- A contact (when the member is unknown in the CID)

Each assignment is referenced by an assignment number (legacy reference). The assignment number is unique to a specific trouble ticket.

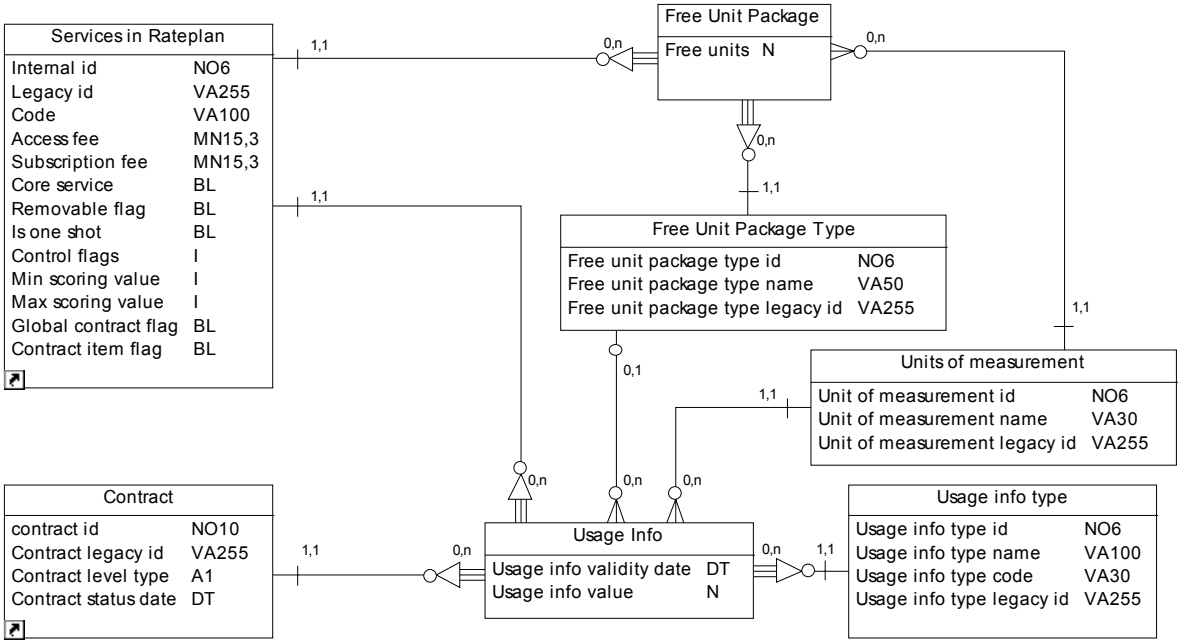
The assignment is also defined by the date of the assignment and a note field that can be updated by the currently assigned user.

Trouble ticket problem types can be grouped by category.

Available trouble ticket problem types depend on the organization type.

USAGE INFORMATION

Submodel Diagram



About this Submodel

This submodel defines usage information for a contract.

Usage Info is defined by:

- A value
- A validity date of the value
- The related service
- The usage info type (prepaid credit, number of units used in a package, number of units outside the package, number of international unbilled communication units…)
- The value unit (minute, money, …)
- If consistent, the related free unit package

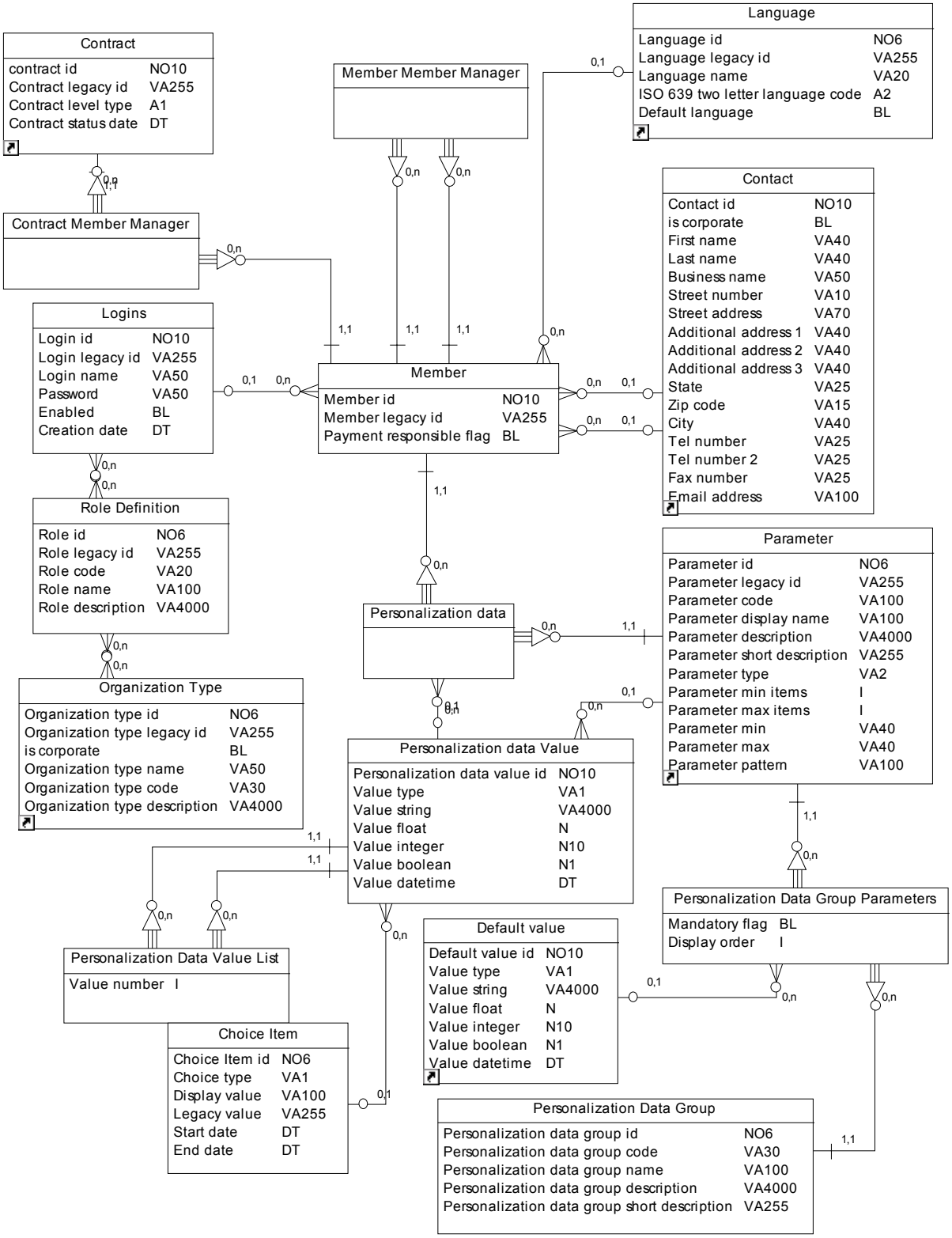
For prepaid contracts, the usage info is not related to a free unit package.

In case of a rateplan without any services, a fake service must be declared in order to link the usage info to it.

There cannot be more than one usage info for a trio (Contract, ServiceInRatePlanInfo, UsageInfoType)

USER

Submodel Diagram


Contract
contract id NO10
Contract legacy id VA255
Contract level type A1
Contract status date DT
Member Member Manager
Language
Language id NO6
Language legacy id VA255
Language name VA20
ISO 639 two letter language code A2
Default language BL
Contact
Contact id NO10
is corporate BL
First name VA40
Last name VA40
Business name VA50
Street number VA10
Street address VA70
Additional address 1 VA40
Additional address 2 VA40
Additional address 3 VA40
State VA25
Zip code VA15
City VA40
Tel number VA25
Tel number 2 VA25
Fax number VA25
Email address VA100
Contract Member Manager
Logins
Login id NO10
Login legacy id VA255
Login name VA50
Password VA50
Enabled BL
Creation date DT
Member
Member id NO10
Member legacy id VA255
Payment responsible flag BL
Role Definition
Role id NO6
Role legacy id VA255
Role code VA20
Role name VA100
Role description VA4000
Personalization data
Parameter
Parameter id NO6
Parameter legacy id VA255
Parameter code VA100
Parameter display name VA100
Parameter description VA4000
Parameter short description VA255
Parameter type VA2
Parameter min items I
Parameter max items I
Parameter min VA40
Parameter max VA40
Parameter pattern VA100
Organization Type
Organization type id NO6
Organization type legacy id VA255
is corporate BL
Organization type name VA50
Organization type code VA30
Organization type description VA4000
Personalization data Value
Personalization data value id NO10
Value type VA1
Value string VA4000
Value float N
Value integer N10
Value boolean N1
Value datetime DT
Personalization Data Group Parameters
Mandatory flag BL
Display order I
Personalization Data Value List
Value number I
Default value
Default value id NO10
Value type VA1
Value string VA4000
Value float N
Value integer N10
Value boolean N1
Value datetime DT
Choice Item
Choice Item id NO6
Choice type VA1
Display value VA100
Legacy value VA255
Start date DT
End date DT
Personalization Data Group
Personalization data group id NO6
Personalization data group code VA30
Personalization data group name VA100
Personalization data group description VA4000
Personalization data group short description VA255

About this Package

This submodel defines users. A user is a physical person belonging to an organization.

A user is defined by:

- A legal contact
- A billing contact
- A language
- A login
- A password

The role(s) assigned to him or her

A role defines the features available to a user, and the scope of each feature. The list of possible roles for an individual depends on his or her organization type. A user can have more than one role. In this case, less restrictive rules apply.

If you create a new customized role, the value of the ID must be greater than 100.

Role definitions (role name and description) are stored in the CID Role Definition table. The security features associated with a role (the list of allowed actions) are stored in the BLM `security.xml` customization file.

Additional Details

Passwords are stored following the RFC2307. Their value must be represented using the
following syntax:

```
passwordvalue = [schemeprefix] encryptedpassword
   schemeprefix = "{" scheme "}" if the password is encrypted
scheme = "MD5" / "SHA"
encryptedpassword = encrypted password
```

The request from the database converts login names to lower case. Therefore only lower case login names can be used in the database.

Personalization Data

Every user can be qualified by personalization data. Personalization data is a set of values specified by parameter descriptors. The different personalization data parameters are grouped in personalization data groups.

You can manage these groups directly from the presentation layer, using the personalization data group code. Personalization data groups represent a functionally consistent set of parameters. A parameter can only be in one group.

You can define a default value for every personalization data parameter.

CHAPTER 4

Sample Data

In This Section

About the Sample Data..... 86
Creating the CID Database..... 87
Contents of the Sample Data..... 89

About the Sample Data

In order for you channels to run correctly, you need to have information in the CID that the application can access and manage. The CID comes with sample data that you use while working with channels. Sample data include reference data (countries, and so on), a product catalog, customer/dealer/customer service representatives (CSR) data and user data.

The comprehensive sample data help you build and test your application quickly and easily as you do not need to create test data.

When installing the CID for development or demonstrations, you can use the `cidAdminTool` to:

- Install the CID with system data only

  When you install the CID with system data only, you can then populate it with your own sample data.
- Install the CID with system and sample data

  When installing the CID with both types of data, the CID is ready to be used for development or demonstrations.

Creating the CID Database

To create the CID and populate it with sample data for the Personalization Manager channels, you use the cidAdmin tool.

When working with the demo CID Database, server components can only interact with the database using the <CID_ADMIN> account.

To create the CID with system data only

1 Go to <home_dir>/bin.

2 Run the CID Administration tool. Use the syntax:

```
cidAdminTool create_demo_cid_structure <CID> <CID_ADMIN login>
<CID_ADMIN password>
```

where <CID>:

- Oracle: <instance alias>
- DB2: <database alias>
- SQL Server: <database host> [:<port>] If no port is specified, the tool uses the default SQL server port

When finished, the CID Administration tool displays a message.

3 Populate the CID with your sample data. You can use the sample data delivered with the TSM. This data is located in <home_dir>/data/cid/<your_dbms>/data.

When working with the demo CID Database, server components can only interact with the database using the <CID_ADMIN> account.

To create the CID with system and sample data

1 Go to <home_dir>/bin.

2 Run the CID Administration tool. Use the syntax:

```
cidAdminTool create_demo_cid_test <CID> <CID_ADMIN login>
<CID_ADMIN password>
```

where <CID>:

- Oracle: <instance alias>
- DB2: <database alias>
- SQL Server: <database host> [:<port>] If no port is specified, the tool uses the default SQL server port

When finished, the CID Administration tool displays a message.

To remove the CID

1. Go to <home_dir>/bin.
2. 2. Run the CID Administration tool. Use the syntax:

   cidAdminTool drop_demo_cid_structure <CID> <CID_ADMIN login> <CID_ADMIN password>

   where <CID>:

   - Oracle: <instance alias>
   - DB2: <database alias>
   - SQL Server: <database host> [:<port>] If no port is specified, the tool uses the default SQL server port

   When finished, the CID Administration tool displays a message.

Contents of the Sample Data

Reference Data

Along with the system data, the sample data includes reference data. In the CID, the reference data includes lists of languages, countries, and other general data.

Product Catalog Data

Corresponding to a Communication Service Providers catalog, the sample data includes service data which includes services, commercial offers, rate plans and other data.

User and Associated Customer/Dealer/CSR Information

The sample data includes a set of users and user types. This set of data includes the following logins:

The password is the same as the login.

When logging in, you must enter the login and password in lowercase.

LOGIN	ACT	NOTES	CHANNELS
joe	Consumer administrator	Has 1 contract	WEB WAP
tammy	Consumer administrator	Has more than 1 contract	WEB WAP
0660100032	Consumer administrator	Prepaid subscriber	WEB WAP IVR
0660100034	Consumer administrator		WEB WAP IVR
admadm	Business administrator	Administrator of Acme corporation	WEB
adm1, adm2, michel	Business administrator	Administrators of sub-levels of Acme corporation	WEB
contractadm	Contract administrator	Manages a set of Acme corporation contracts	WEB

LOGIN	ACT	NOTES	CHANNELS
bigboss	Business user	User at the top level of Acme corporation Has more than 1 contract	WEB
véro, jean, paul, victor	Business user	User at a sub-level of Acme corporation	WEB
jack	Business administrator	Administrator of Jack and Co	WEB
jim	Business user	User at the top level of Jack and Co	WEB
tph	Dealer user	User at the top level of dealer organization	WEB
herve	Dealer user	User at a sub-level of dealer organization	WEB
eur	Telco user		WEB
will	Supplier user		WEB
acctmgr	Telco business account manager	Initially in charge of Acme corporation (list of managed business customers can be changed by senior_acctmgr)	WEB
acctmgr2	Telco business account manager	Initially in charge of Jack and Co (list of managed customers can be changed by senior_acctmgr)	WEB
senior_acctmgr	Telco business senior account manager	Manages all business accounts	WEB

CHAPTER 5

Customizing the CID

In This Section

About Customizing the CID 92
Reserved Tables..... 93
Modifying Tables and Records 94

About Customizing the CID

You can modify some of the reference information in the CID.

When modifying the information in the CID:

- Do not modify the information in reserved tables. The information in these tables is system information.
- Follow the rules when modifying tables containing reference information.
- Legacy IDs can be changed in all tables.

Except for the reserved tables and the restrictions outlined in this section, you can modify the contents of the CID.

Reserved Tables

The following tables are reference data. You must not modify them.

TABLE	NOTES
ACTION	Do not modify
APP	Do not modify
OBJECT_TYPE	Do not modify
REQUEST_STATUS	Do not modify
RULE_REF	Do not modify
NOTIFICATION_TYPE	Do not modify
NOTIFICATION_STATUS	Do not modify

Modifying Tables and Records

The following table describes the reserved ranges and modification rules:

TABLE	NOTES
ACTION_PRIORITY	You can modify existing information. Do not add new records.
AVL_APP_LANG	You can modify existing information. You can add new records.
CHOICE_ITEM	Do not modify CHOICES with a CHOICE_ITEM_ID less than 1000 Your choice items must have a CHOICE_ITEM_ID equal to, or greater than 1000, and a PARAM_ID equal to, or greater than 1000
COMPOSITE_PARAMETER	Do not modify COMPOSITE PARAMETERS with a PARAM_ID less than 1000 Your composite parameters must have a PARAM_ID equal to, or greater than 1000
COUNTRY	Your COUNTRY_ID must be greater than 1000 The sample data contain ISO-3166 country names.
CURRENCY	Your CURRENCY_ID must be greater than 1000 The sample data contain ISO-4217 currency codes.
LANGUAGE	Do not modify LANGUAGES with a LANG_ID=0 Your languages must have a LANG_ID greater than 1000 French must have a LANG_CODE='fr' English must have a LANG_CODE='en' The sample data contain ISO-639 languages codes.
LOGINS	Do not modify LOGINS with a LOGIN_ID less than 10 Your logins must have a LOGIN_ID greater than 1000 The sequence generating LOGIN_IDs starts at 1000
MEMBER	Do not modify MEMBERS with a MEMBER_ID less than 10 Your members must have a MEMBER_ID greater than 1000 The sequence generating MEMBER_IDs starts at 1000
ORGANIZATION	Do not modify ORGANIZATIONS with an ORGANIZATION_ID less than 10 Your organizations must have an ORGANIZATION_ID greater than 10 The sequence generating ORGANIZATION_IDs starts at 1000
ORGANIZATION_TYPE and ORG_TYPE_ROLE_LINK	Do not modify ORGANIZATION TYPES with an ORG_TYPE_ID=0 Your organization types must have an ORG_TYPE_ID greater than 0
ORGANIZATION_TYPE_CATEGORY	Do not modify ORGANIZATION TYPES with an ORG_TYPE_CATEGORY_ID=0 Your organization type categories must have an ORG_TYPE_CATEGORY_ID greater than 0

TABLE	NOTES
PARAMETER	Do not modify PARAMETERS with a PARAM_ID less than 1000 Your parameters must have a PARAM_ID equal to, or greater than 1000
PRIORITIES	Do not add or remove records You can modify the values in PRIORITY_DISTRIB. The total of the values in PRIORITY_DISTRIB must be equal to 100
ROLE_DEF	Do not modify ROLES with a ROLE_ID less than 10 Your roles must have a ROLE_ID equal to, or greater than 10
SEARCH_FILTER	You can modify the following values of existing filters: ROW_COUNT IS_DEFAULT START_DATE END_DATE Your filters must have a SEARCH_FILTER_ID equal to, or greater than 1000
SEARCH_CRITERIA	You can modify the following values of existing criteria: • IS_ACTIVE • IS_MANDATORY • DISPLAY_ORDER Your criteria must have a PARAM_ID equal to, or greater than 1000
UNIVERSAL_STRING	Translations with a STRING_ID less than 20000 are used for the translation of reference data. Do not translate the system reference data directly in the tables, but use this table instead. STRING_IDs under 2000 are subject to change without notice and you may lose any modification you make. You may change the translations or add translations in other languages as long as you do not change the STRING_IDs. Your data translations must have a STRING_ID equal to, or greater than 20000.
USER_TYPE_EVENT	For a USER_TYPE_EVENT_ID less than 1000, you may only change the ACTIVATION_FLAG Your events must have a USER_TYPE_EVENT_ID equal to, or greater than 1000

Index

A

ACTION CID Submodel
- about • 22
- adding additional parameters • 24
- diagram • 21
- IS_REMOVABLE flag • 22
- IS_REPLACEABLE flag • 22

Actions
- adding additional parameters to • 24
- and the CID • 22
- IS_REMOVABLE flag • 22
- IS_REPLACEABLE flag • 22
- types • 22

Additional Parameters
- in a request • 24
- in actions • 22
- setting rules for • 22

AUDIT Submodel
- about • 27
- and DO events • 27
- and User Events • 27
- diagram • 26

B

BILLING ACCOUNT CID Submodel
- about • 31
- diagram • 30

C

CID (Customer Interaction Datastore)
- about • 16
- adding sample data • 87
- customizing • 92
- removing sample data • 88
- reserved tables and data • 93, 94
- submodels • 19

cidAdminTool Administration Tool
- create_demo_cid_test command • 87
- drop_demo_cid_structure command • 88

COMMERCIAL OFFER CID Submodel
- about • 33
- diagram • 32

CONTACT CID Submodel
- about • 35
- diagram • 35

CONTRACT CID Submodel
- about • 38
- diagram • 36

Customizing
- CID • 92

D

DO Events
- and the AUDIT submodel • 27

DOCUMENTATION CID Submodel
- about • 39
- diagram • 39

F

FILTER CID Submodel
- about • 41
- diagram • 40

H

Help
- technical support • ix

I

INVOICE CID Submodel
- about • 43
- diagram • 42

N

NOTIFICATION CID Submodel
- about • 45
- diagram • 44

O

ORGANIZATION CID Submodel
- about • 47
- diagram • 46

ORGANIZATION VIEW CID Submodel

about • 50
diagram • 49

P

PARAMETER CID Submodel
about • 52
diagram • 51
Parameters
in the CID • 52
storing • 55, 56, 57
types • 52
values • 55
PAYMENT CID Submodel
about • 58
diagram • 58
PAYMENT METHOD CID Submodel
about • 59
diagram • 59
Persistent Action Managers
in the CID • 70, 71
Personalization Data
in the CID • 80, 83
Personalization Manager
sample data • 86, 87, 89
PROCUREMENT CID Submodel
about • 61
and approval processes • 61
diagram • 60

R

RATEPLAN CID Submodel
about • 63
diagram • 62
REQUEST CID Submodel
about • 66
diagram • 65
Requests
additional parameters • 22
in the CID • 65, 66

S

Sample Data
about • 86, 87
contents • 89
inserting into CID • 87
reference • 89
removing from CID • 88
Sample Data - services • 89
users • 89
SERVICE CID Submodel
about • 69
diagram • 68
SHOPPING CART CID Submodel
about • 71
diagram • 70
Shopping Carts
in the CID • 70, 71
SYSTEM CID Submodel
about • 73
and object types • 74
diagram • 73

T

TROUBLE TICKET CID Submodel
about • 77
diagram • 75

U

USAGE INFORMATION CID Submodel
about • 79
diagram • 78
USER CID Submodel
about • 82
and Personalization Data • 83
diagram • 80