edocs®

CBU Reference Guide

V4.0
Document ID: TAGN-07-4.0-01
Date Published: 9.10.03

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

Preface

In This Section

Using this Manual

Welcome to the Communications Billing and Usage (CBU) Reference Guide.

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

Before You Get Started

You should be familiar with the following:

- Your application architecture
- The CBU
- Designing or working with databases and data warehouses
- SQL

Who Should Read this Manual

This manual is a reference guide for the CBU. This guide is for anybody who needs information about content and structure of the CBU.

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

- Administrators

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

- Developers

  This guide contains the conceptual structure of the CBU you can use to understand how the CBU organizes information. You need to be familiar with the tables and their relationships. There is also important information about loading the CBU.

- Project Architect

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

- Project Manager

  As this reference guide contains information about the CBU 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 contains the following chapters:

- **Overview of the CBU**

  This chapter covers the role and content of the CBU.

  It contains information about:

  - The design of the CBU Tables
  - The Design conventions

- **CBU Model**

  This chapter covers the submodels of the CBU.

  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

- **Loading the CBU**

  This chapter covers loading information into the CBU.

  It contains information about:

  - The loading policy
  - Loading Dimensions
  - Loading Fact Tables
  - Loading Security Tables

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

edocs Telco Solutions comes with comprehensive documentation that covers all aspects of building TAM solutions. You should always read the release bulletin for late-breaking information.

Getting Started

If you are new to edocs Telco Solutions, you should start by reading *Introducing Telco Analytics 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 that covers different aspects of working with the application. At the beginning of each manual, you will find an introductory chapter that covers concepts and tasks.

Designing Your Solution

While reading *Introducing Telco Analytics Manager*, you can begin to envision how the different components can address your solution's needs.

You can refer to *Developing Analytical Applications* for information about customizing the database, synchronizing data with Telco Service Manager (TSM), loading data from external invoice files, and other design issues. The *CBU Reference Guide* also gives you the information about how the information in your solution is managed and stored. You should also read the section on integrating Telco Analytics Manager with Telco Service Manager in *Developing Analytical Applications*.

Installing Your Analytical Application

You should start by reading the Release Bulletin. For detailed installation and configuring information, refer to *Installing Telco Analytics Manager*. This manual covers installing *Telco Analytics Manager* on one or more computers. It also contains the information you need to configure the different components you install.

Building Analytical Solutions

If you are designing and programming analytical applications, you have several different sources of information. If you are programming the user interface of the solution, you should read *Building Reports*. You can also refer to the *QRA API Specification* and the *QRA Configuration File Reference Documentation* for detailed information about the different components you can use to build reports which serve as the user interface. For configuring the various components, refer to *Installing Telco Analytics Manager* and sections in other documents that specifically deal with the component to be configured.

If you are working on the data warehouse side of TAM and are interested in how the information will be loaded into the data warehouse, you should read *Developing Analytical Applications*. For more information about the design and structure of the CBU, you should refer to the *CBU Reference Guide* along with the *CBU Reference* documentation for your database.

Integrating TAM and TSM

If you are involved in configuring your solution to work with Telco Service Manager, you should read *Introducing Telco Analytics Manager* for an overview of the components and how they interact. You should then read *Developing Analytical Applications* for information about synchronizing data between the Telco Analytics Manager and Telco Service Manager. In this manual, you will also find information about loading data in both the CBU and the Telco Service Manager.

Managing Telco Analytics Manager

If you are responsible for managing Telco Analytics Manager, you should read the *Installing Telco Analytics Manager* for information about configuring various components. *Administrating Telco Analytics Manager* covers what you need to know about managing your solution at runtime.

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

CHAPTER 1

Overview of the CBU

In This Section

About the CBU

The CBU database contains customers billing and usage information.

Users can analyze their invoices using:

- Reference dimensions such as date/time, tariff, and service
- Customer-specific information such as organization views, contracts, and billing accounts

The CBU also comes with built-in security based on roles and scopes. This type of security, which mirrors the security of your Account Management application, ensures the security and confidentiality of the information in the CBU. User access to the CBU depends on user authentication and the user's assigned roles and scopes.

The CBU is designed to allow users to analyze their own invoices and usage. It is not designed for analysis of the entire customer base.

For more information about:

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

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

- Sample data, refer to the CBU demokit readme documentation. This documentation is located in `<home_dir>/samples/cbu/demokit` in a subdirectory corresponding to your database.

About the CBU Tables

The CBU contains the following types of tables:

- Fact tables
- Dimension tables
- Security tables
- System tables

Fact Tables

A fact table contains raw information and is the central table of the CBU. These tables contain the information users want to analyze, such as invoices and usage details.

Dimension Tables

To analyze the raw information in the fact tables, you use dimensions. A dimension is a category of information along with a set of attributes. For example, one of the most common dimensions is time, which you can use to filter the contents of a fact table.

The CBU has the following dimensions:

- **Reference dimensions**

  These dimensions are predefined reference dimensions such as date and time dimensions along with bill periods and other business references.

  For example, when building reports of bill usage detail records in the `NMY_BUDR_FACT` table, you can use the Date and Time dimension to restrict the time period, then apply the Bill Period dimension. Once you have obtained this result, you can then apply a Business Reference such as the service you want in your report.

  Reference dimensions include:

  - Date and Time
  - Bill Period

  The bill period is a period of time that is used to calculate the amount due for services. A bill period has a name and is associated with a bill cycle. A bill period may have a start and end date.

  For instance, a bill period may correspond to a calendar month (January, February, and so on) or may be shorter or longer than that (first half of a month, a day for one time events, and so on.)

- **Business Reference dimensions**

  Business Reference Dimensions are reference dimensions that correspond to information specific to a Telecommunication Service Provider as destination zone, usage type, and so on.

- **Customer data dimensions**

  Customer Data dimensions are dimensions that use information directly associated with a customer. Unlike reference information, which is relatively static, customer data dimensions change and are updated frequently. You use these dimensions to extract information from the fact table that directly concerns the user for your reports.

  For example, when building reports of bill usage detail records in the `NMY_BUDR_FACT` table, you can use the Date and Time dimension to restrict the time period, and then apply the Contract dimension. This allows you to quickly obtain a report for a specific contract.

  Customer Data dimensions include:

  - Contract
  - Billing Account
  - Cost Center

Security Tables

Security tables contain information relative to access permission.

Security tables include:

- Users
- Actors
- ACL (Access Control List)

Security is handled by affecting roles to users.

A role combined with user scopes define the general access rules of objects. Scopes determine what objects can be accessed according to the user’s member and organization level information.

The available scopes:

- Intra-organization scopes
  - All objects of the organization hierarchy
  - All objects related to the member's level or below
  - All objects owned by the member including associated sub-objects
  - All objects explicitly managed by the member (limited to contracts and members)
- Inter-organization scopes
  - All organizations
  - All organizations managed by the member's level
  - All organizations managed by the top level of the member's organization
  - All organizations managed by the member

All intra-organization scopes can be directly managed in CBU.

For more information on roles, scopes and security, refer to the *BLM Reference Guide*.

There are two methods of managing secure access to CBU objects:

- CBU based security

  You can ensure user confidentiality by implementing security directly in the report SQL query. This query is based on the security tables in the CBU.

  When using this method, the presentation layer must only authenticate the user by using the BLM API or other supported authentication methods.

  You can only use this method for intra-organization scopes and if users are synchronized with in the CBU.

  For more information about how this works, refer to *Creating Reports using SQL* in this manual.

- QRA based security

  You can use the QRA components to manage security by using system prompts. When using this method, you write SQL queries directly accessing the customer dimension table

  For more information about how this works, refer to *Developing Reports*.

System Tables

System Tables contain system information for the CBU or applications accessing the CBU.

CBU Design Conventions

Before you start, it is important to introduce the conventions used by the CBU. This way, when you see diagrams and names of tables, you can easily recognize the contents and use of the different tables.

In the CBU, the contents adhere to the following conventions:

Tables

- Start with the single prefix `NMY_`
- Have a name composed of words in caps letters separated by underscore.
- End with normalized suffixes:
  - `_FACT` Fact tables
  - `_DIM` Dimensions and security tables
  - `_LNK` Link tables
  - `_ACL` Access control list tables
  - `_HLP` Helper tables
  - `_INFO` Information tables

Table Keys

All table key columns names start with the table name (without prefix and suffix).

- Surrogate Key: Internal primary key of a dimension table and always end with suffix `_KEY`
- Business Key: Backend system identifier of a dimension table and always end with suffix `_BIZ`

  Certain business keys can be null.

Attribute Names

To make the CBU model easier to read and understand, standard logical types have been defined to highlight the characteristics and the semantic of attributes. Whenever possible, the attribute naming has also been standardized to enhance the comprehension of the CBU physical model.

The following table describes the following for each the logical types:

- Type
- Description
- Related attribute suffix
- Example of their physical implementation (for Oracle)

LOGICAL TYPE	DESCRIPTION	SUFFIX	PHYSICAL IMPLEMENTATIO N (ORACLE CONVENTIONS)
T_ACCID	Network ACCess Identifier (example: destination number)	No rule	VARCHAR2(50)
T_BIZ	Nullable business key	_BIZ	VARCHAR2(255)
T_BIZ_AK	Non Nullable business key	_BIZ	VARCHAR2(255)
T_CODE	Attribute storing a value corresponding to a codification	_CODE	VARCHAR2(32)
T_DESC	Any textual attribute fully describing a record.	_DESC	VARCHAR2(255)
T_FLAG	Any boolean attribute	_FLAG	NUMBER(1)
T_INT2	Any integer attribute with a maximum value of 2 digits	No rule	NUMBER(2)
T_INT4	Any integer attribute with a maximum value of 4 digits	No rule	NUMBER(4)
T_INT6	Any integer attribute with a maximum value of 6 digits	No rule	NUMBER(4)
T_KEY	Surrogate key of a dimension table	_KEY	NUMBER(18)
T_KEY_DT	Surrogate key of date dimension table	_DATE_KEY	NUMBER(8)
T_KEY_TM	Surrogate key of time dimension table	_TIME_KEY	NUMBER(4)
T_LABEL	Any textual attribute that can be used to display a record and to select it	_NAME	VARCHAR2(64)
T_MONEY	Any numeric attribute storing amounts value	No rule	NUMERIC(15,3)
T_NAME	Any textual attribute used to display a record	_NAME	VARCHAR2(128)

LOGICAL TYPE	DESCRIPTION	SUFFIX	PHYSICAL IMPLEMENTATION (ORACLE CONVENTIONS)
T_PATH	Complete hierarchical path name of a record (example: cost center path name)	_PATH	VARCHAR2(1024)
T_QTY	Any integer attribute storing a volume	_VOLUME	INTEGER
T_UIDN	Unique ID Number Unique identifier shared by both the end user AND the backend system. Can be null.	_UIDN	VARCHAR2(64)
T_UIDN_AK	Same as T_UIDN but cannot be null	_UIDN	VARCHAR2(64)

CHAPTER 2

CBU Model

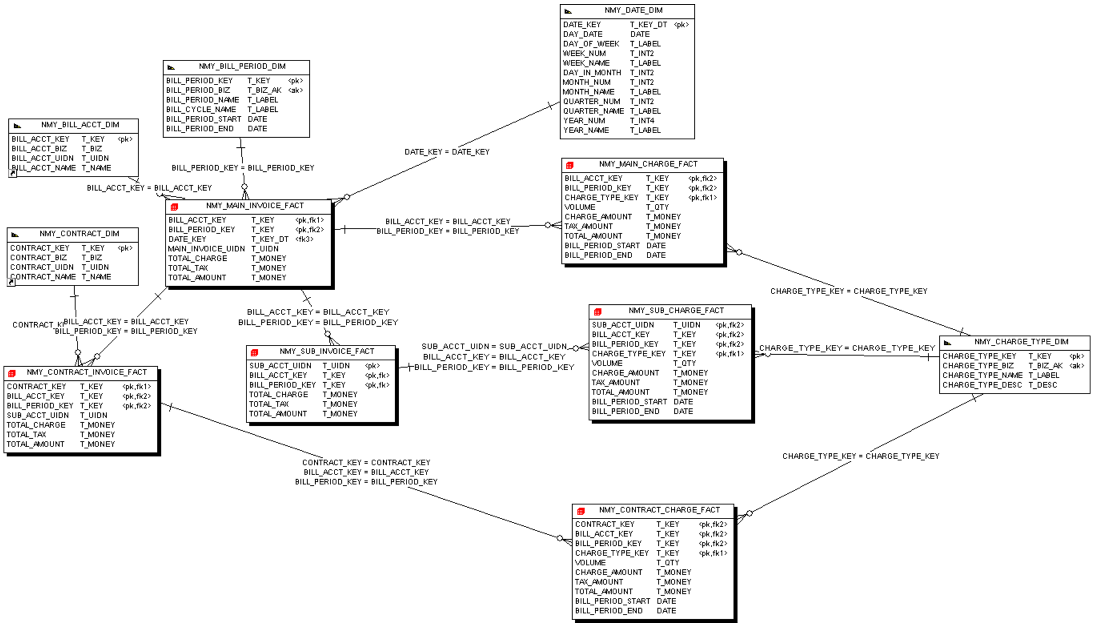
In This Section

About the CBU Model

The CBU model is divided into the following sub models:

- The Invoices sub model
- The Billed Usage Detail Records sub model
- The Cost Center sub model
- The Security sub model
- The System sub model

Invoices Sub Model



The CBU invoices sub model specifies the following tables:

- Fact tables
  - `NMY_MAIN_INVOICE_FACT`

    This fact table contains main invoice records. A main invoice is uniquely identified by its billing account and a bill period.

    The invoice number (`MAIN_INVOICE_UIDN`) is optional.
  - `NMY_SUB_INVOICE_FACT`

    This fact table contains sub invoice records. A sub invoice is a decomposition of a main invoice for a sub set of contracts billed on the main invoice
    The `SUB_ACCT_UIDN` attribute allows referencing the sub account related to the sub invoice without requiring its storage.
  - `NMY_CONTRACT_INVOICE_FACT`

    This fact table contains contract invoice records. A contract invoice is the detail of a main invoice for a specific contract billed on the main invoice.
    The `SUB_ACCT_UIDN` attribute allows storing the link between the contract invoice and the associated sub invoice if any.
  - `NMY_MAIN_CHARGE_FACT`

    This fact table contains the detail of every charge of a specific main invoice.
    An amount value and a volume can be specified for every charge. The real billed period of the charge can also be specified for display purpose.
  - `NMY_SUB_CHARGE_FACT`

This fact table contains the detail of every charge of a specific sub invoice.

- `NMY_CONTRACT_CHARGE_FACT`

  This fact table contains the detail of every charge of a specific contract invoice.

- Dimension tables
  - `NMY_CONTRACT_DIM`

    This dimension table contains contracts.

  - `NMY_BILL_ACCT_DIM`

    This dimension table contains billing accounts.

  - `NMY_BILL_PERIOD_DIM`

    This dimension table contains the bill periods. A record must be created for every bill run.

  - `NMY_DATE_DIM`

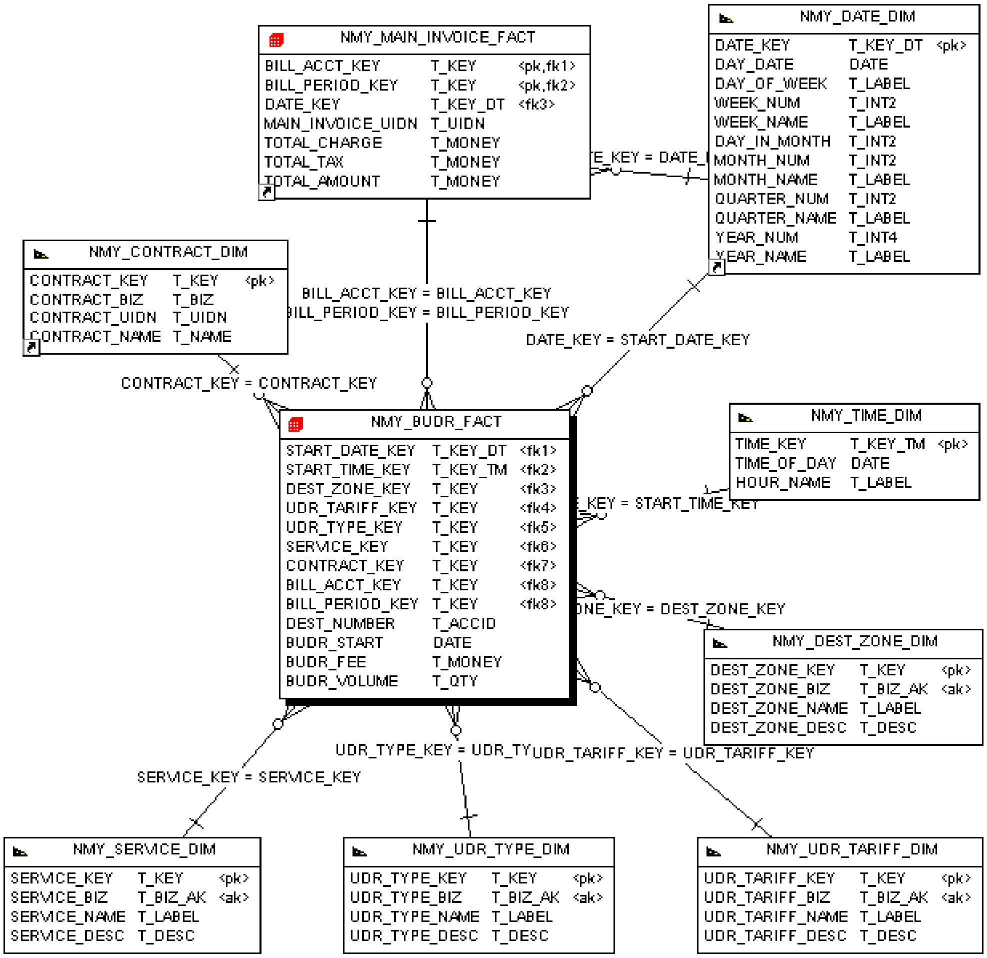
    This dimension table contains dates.

  - `NMY_CHARGE_TYPE_DIM`

    This dimension table contains charge types. A charge type must be specified for every charge.

By default, the model does not manage charge types hierarchy. If required, refer to *Managing Charge Type Hierarchies* in *Developing Analytical Applications*.

Billed Usage Detail Records Sub Model



The CBU billed and usage detail records sub model specifies the following tables:

- Fact tables
  - `NMY_BUDR_FACT`

    This fact table contains billed usage detail records.
  - `NMY_MAIN_INVOICE_FACT`

    This fact table contains main invoice records.
- Dimension tables
  - `NMY_TIME_DIM`

    This dimension table contains time periods.
  - `NMY_DEST_ZONE_DIM`

This dimension table contains destination zone.

- `NMY_UDR_TARIFF_DIM`

  This dimension table contains the tariffs of Usage Detail Records.

- `NMY_UDR_TYPE_DIM`

  This dimension table contains the types of Usage Detail Record types.

- `NMY_SERVICE_DIM`

  This dimension table contains the services.

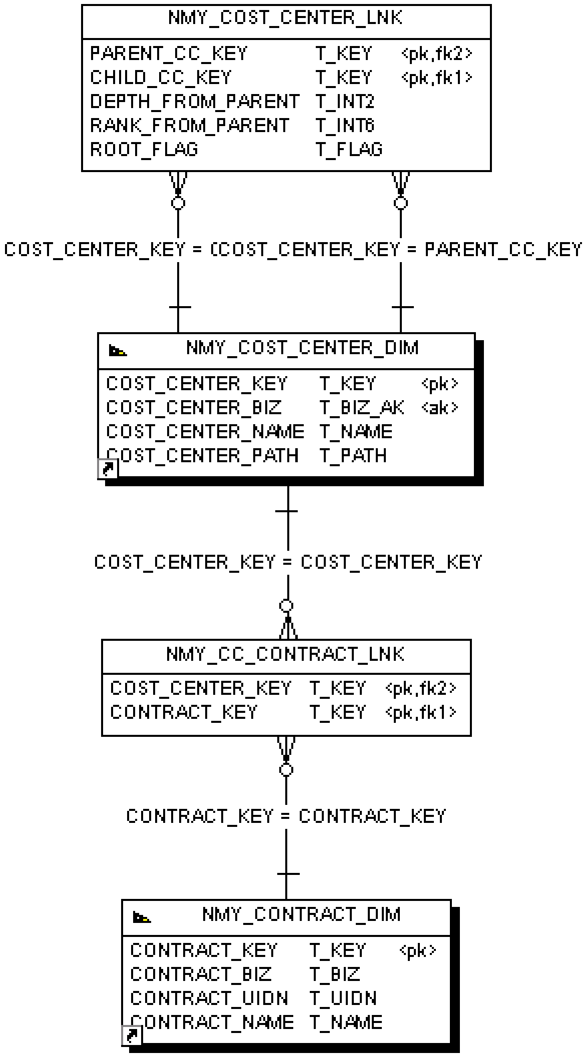
- `NMY_DATE_DIM`

  This dimension table contains dates

- `NMY_CONTRACT_DIM`

  This dimension table contains contracts.

Cost Centers Sub Model


NMY_COST_CENTER_LNK
PARENT_CC_KEY T_KEY <pk,fk2>
CHILD_CC_KEY T_KEY <pk,fk1>
DEPTH_FROM_PARENT T_INT2
RANK_FROM_PARENT T_INT6
ROOT_FLAG T_FLAG
COST_CENTER_KEY = CCOST_CENTER_KEY = PARENT_CC_KEY
NMY_COST_CENTER_DIM
COST_CENTER_KEY T_KEY <pk>
COST_CENTER_BIZ T_BIZ_AK <ak>
COST_CENTER_NAME T_NAME
COST_CENTER_PATH T_PATH
COST_CENTER_KEY = COST_CENTER_KEY
NMY_CC_CONTRACT_LNK
COST_CENTER_KEY T_KEY <pk,fk2>
CONTRACT_KEY T_KEY <pk,fk1>
CONTRACT_KEY = CONTRACT_KEY
NMY_CONTRACT_DIM
CONTRACT_KEY T_KEY <pk>
CONTRACT_BIZ T_BIZ
CONTRACT_UIDN T_UIDN
CONTRACT_NAME T_NAME

The CBU Cost Centers sub model specifies the following tables:

- Dimension tables
  - `NMY_COST_CENTER_DIM`

    This dimension table contains cost centers
  - `NMY_CONTRACT_DIM`

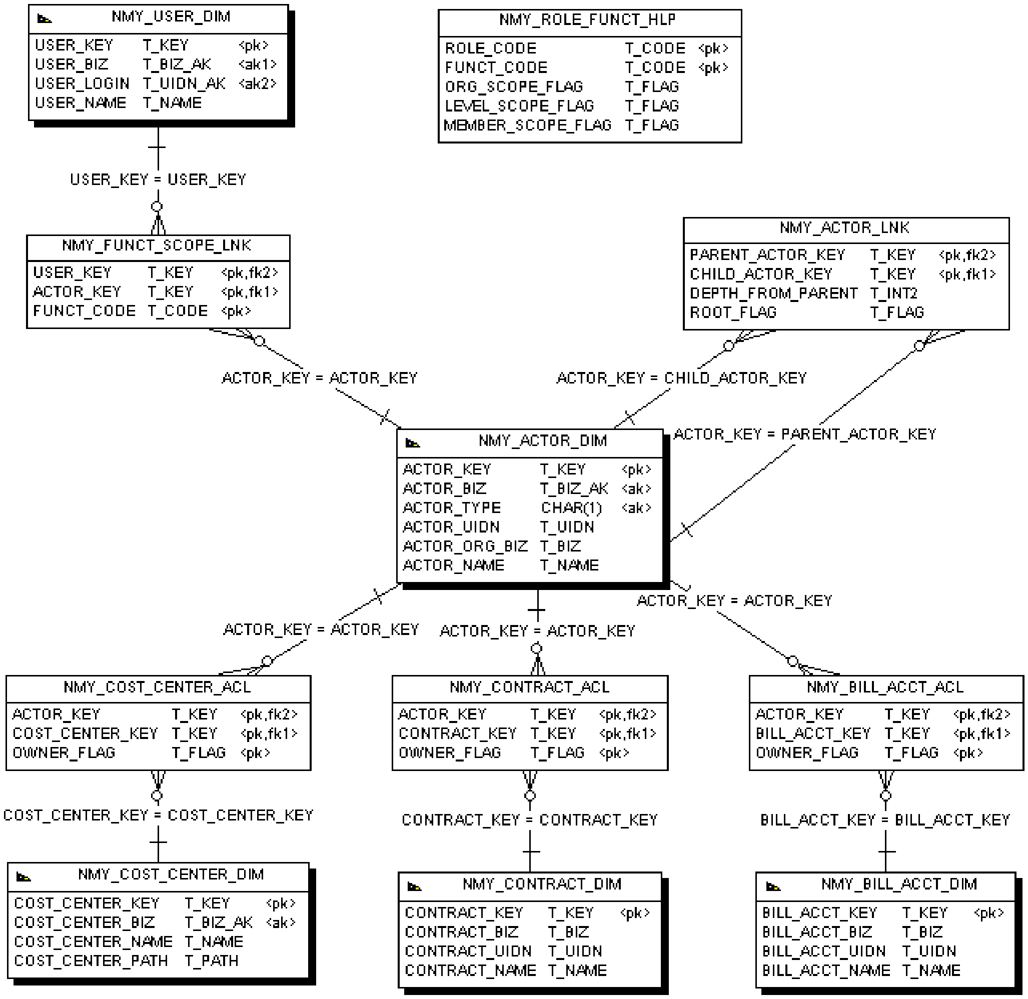
    This dimension table contains contracts.
- Link tables
  - `NMY_COST_CENTER_LNK`

    This dimension table contains links between cost centers to store the cost center hierarchy.

- `NMY_CC_CONTRACT_LNK`

  This dimension table contains links between cost centers and contracts to store the contract grouping inside cost centers.

Security Sub Model



To handle security, the CBU uses the following tables:

- **User table** (NMY_USER_DIM)

  The user table stores the users identities of every user accessing the CBU. This table is not provided to perform user authentication. Users must be authenticated by the presentation layer prior to accessing the CBU.

  Feeding this table is only necessary if you want to implement security directly in report SQL queries.

- **Functional Scope link table** (NMY_FUNCT_SCOPE_LNK)

  This table specifies the scope of the user for each reporting function. A reporting function is identified by a function code (example: you can define one function code for each fact table).

  The scope specifies the authorized actors for a specific function code.

The following elementary scopes can be handled in CBU

- Member or explicit scope: implemented by the link between the user and the member’s user
- Sub hierarchy scope: implemented by the link between the user and its parent level
- Organization scope: implemented by the link between the user and its user's organization

Its possible to merge elementary scopes for one functional code allowing, for example, a user to have a member and a sub hierarchy scope.

- **Role functional scope helper table** (`NMY_ROLE_FUNCT_HLP`)

  This table specifies the mapping between roles and CBU functional scope. The CID2CBU loader uses this table to create records in the functional scope link table for every user as the following algorithm:

  - `ORG_SCOPE_FLAG = TRUE`: A link must be created, for the functional code, between the user and his organization in the actor table
  - `LEVEL_SCOPE_FLAG = TRUE`: A link must be created, for the functional code, between the user and the user’s member’s parent level
  - `MEMBER_SCOPE_FLAG = TRUE`: A link must be created, for the functional code, between the user and his member.

  If you want to implement security directly in SQL report queries, you must first initialize this table with roles and functional scope you want to manage before being able to synchronize CID users in the CBU.

- **Actor and Actor link tables** (`NMY_ACTOR_DIM` and `NMY_ACTOR_LNK`)

  An actor gives access to a specific list of resources. An actor can be an organization, a level or a member.

  - The `NMY_ACTOR_LNK` table specifies the organization hierarchy. It contains every hierarchy links between parent level (including organization as root level) and all sub hierarchy levels.
  - Member must not be linked to their parent level.

- **ACL tables**

  These tables control access of an actor on the following resources:

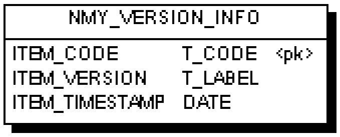
  - Billing Accounts
  - Contracts
  - Cost Centers

  The Owner flag in ACL tables allow you to differentiate between objects owned by an actor or explicitly managed by the actor (for example, managing the difference between the owner of a contract and a manager of a contract)

Secured Objects

- Customer data dimensions: The ACL tables with actors secure them.
- Fact tables: There is no direct ACL on fact tables. Relevant customer data dimensions secure them.
  Billing account dimension secures the Main invoice fact table.
  Contract dimension secures the Contract Invoice and Billed Usage Detail Record fact tables
- Time and business reference dimensions: There is no security on these tables.

System Sub Model



NMY_VERSION_INFO		
ITEM_CODE	T_CODE	<pk>
ITEM_VERSION	T_LABEL	
ITEM_TIMESTAMP	DATE	

The CBU system sub model contains the following tables:

- `NMY_VERSION_INFO`

  This table contains the CBU version

CHAPTER 3

Loading the CBU

In This Section

About the Loading Policy

This section outlines the loading policy of the CBU. When you load the CBU, you have to load:

- Customer Dimensions
- Reference Dimensions
- Fact tables
- Security tables

For Customer Dimensions and Security, the CID2CBU loader synchronizes them with the CID. You can also use the CustDim loader to load customer dimensions from invoice files.

You are responsible for loading:

- Fact tables
- Reference Dimensions

Loading Policy and Sequences

Depending on your database, you may need to create sequences or stored procedures when programming the loading of different tables:

- For Oracle, create standard sequence.
- For DB2, create the the `SEQ_<Object_name>_AI_Value` sequence. Be sure to use sequence objects.

  For more information about the sequence objects, refer to the DB2 release notes.
- For SQL Server, create a sequence stored procedure:
  1. Insert a new row in the `KEYSTORAGE` table.

     ```
     insert into KEYSTORAGE(TABLENAME,LASTKEY)
     values('SEQ_<name>_PARAM_VALUE',1000-1)
     ```

     go
  2. Use one of the existing `SP_SEQ_xxxx_AI_VALUE` stored procedure as a template for a creating the new sequence stored procedure `SEQ_<Object_name>_AI_VALUE`

  Example of a stored procedure:

  ```
  insert into KEYSTORAGE(TABLENAME,LASTKEY) values('SEQ_<name>',1000-1)

  go

  create procedure SP_SEQ_<name> as begin

  begin tran

  update KEYSTORAGE set LASTKEY=LASTKEY+1 where TABLENAME='SEQ_<name>'

  if @@rowcount=1 select LASTKEY from KEYSTORAGE where TABLENAME='SEQ_<name>' else select NULL

  commit tran

  end

  go
  ```

Dimension Loading Policy

For Dimension tables, the data in the tables can be:

- Pre-loaded. The entries are loaded into the tables before being used by the fact tables.
- On-demand The entries are marked as being needed by the fact loading process. The loading process creates a new entry with a new surrogate key. The attributes are completed afterwards before the system goes back online.
  - The entry can be reconciled with any other data source by using a biz key
  - Post-processing can fill all attributes
- Synchronized with the CID. The entries a user sees are the same in both the CBU and CID.
- Loaded from invoice files (in both the CBU and CID)

Loading Date and Time Dimensions

Date and Time dimensions are static dimensions and are not synchronized. The values can be pre-calculated and loaded before any other information.

The loading of these dimensions is mandatory and they remain constant for the entire lifespan of the TAM CBU.

Tables:

- `NMY_TIME_DIM`
- `NMY_DATE_DIM`

Loading Guidelines:

1. Define the length of time the TAM CBU will be on duty in years. The tables will have 365/366 records per year. You can load a few decades. The time table always has 1440 (24 x 60) rows.

   Be sure to go far enough in the past to cover the dates of entries in the fact tables.

2. Determine the required attributes for date and time values. *Unused attributes may be omitted (filled with specific constant values to satisfy database constrain*ts).
3. Determine numbering and naming conventions according to your business rules for day, month, quarter, hour, and year.
4. Optionally choose a default/unknown value when date or time information is not provided by facts.
5. Generate and load the data.

Loading Billing Period and Business Reference Dimensions

The Billing Period and Business Reference dimensions can be pre-loaded or loaded on-demand. The contents of this dimension can determine which loading method to use:

- If the content of the dimension does not change often or if you know exactly when it does change (end of a quarter, every night at 3 AM, and so on), use pre-loaded content
- If the content changes frequently or if your application requires up to date information, use on-demand content

The actual loading scenario of these dimensions can be a mix of both solutions and is a compromise between dynamic content and integrity.

Tables:

- `NMY_BILL_PERIOD_DIM`
- `NMY_DEST_ZONE_DIM`
- `NMY_UDR_TARIFF_DIM`
- `NMY_UDR_TYPE_DIM`
- `NMY_SERVICE_DIM`
- `NMY_BILL_PERIOD`

Loading Guidelines for Pre-loaded Dimensions:

1. Determine the biz key algorithm for the dimension. This key will be used to match the fact data stream and the right entry in the dimension. The linking will be made using the surrogate key.
2. Determine the surrogate key algorithm for the dimension.

   This key MUST not encode any information. You can use a sequence or use a line number from dimension data source (a spread sheet for example). For more information about sequences or stored procedures and your database, refer to *Loading Policy and Sequences*.
3. Determine the entries that will be used by facts.
4. It is recommended to choose a default/unknown value when information is not provided by facts.
5. Decide the name for each entry. This name is used in reports to detail or filter facts.
6. Optionally add a description to entries.
7. Generate and load the data.

Loading Guidelines for On-demand Dimensions:

1. Determine the biz key algorithm for the dimension. This key will be used to match the fact data stream and the right entry in the dimension. The linking will be made using the surrogate key.
2. Create a sequence starting at 1 for each dimension table.
3. When loading new facts, if a business key is not found, create a new entry with the given business key using the sequence to generate the surrogate key. Link the fact to it and notify the request to update this key.
    - Decide the name for this new entry.
    - Optionally add a description.
    - Update the notified entries.

Loading Customer Data Dimensions

The volume of the Customer Data dimensions can be constant or slowly growing.

When deploying your Telco Analytics Manager in event mode, you must pre-load this information using the CID2CBU loader. This is due to the sizing requirements. Once loaded, you can implement on-demand loading if required.

When deploying in invoice mode, you can use the CustDim loader to manage loading the customer dimension from invoice files.

For more information about the different modes of deployment, refer to *Introducing Telco Analytics Manager.*

Tables:

- `NMY_CONTRACT_DIM`
- `NMY_BILL_ACCT_DIM`
- `NMY_USER_DIM`
- `NMY_FUNCT_SCOPE_LNK`
- `NMY_ACTOR_DIM`
- `NMY_ACTOR_LNK`
- `NMY_BILL_ACCT_ACL`
- `NMY_CONTRAT_ACL`
- `NMY_COST_CENTER_ACL`

On-Demand Loading Guidelines:

- These guidelines apply to `NMY_CONTRACT_DIM` and `NMY_BILL_ACCT_DIM` tables.
- `NMY_USER_DIM` and `NMY_FUNCT_SCOPE_LNK` can be only synchronized with information in the CID.
- Create a new entry using a sequence when the business key or the UIDN does not match
- Automatically add attributes if possible: name and UIDN may be provided by the data source (fact stream)
- The New entry will be correlated by the CID2CBU loader during synchronization process with the CID.
- You must also initialize the `NMY_ROLE_FUNCT_HLP` table. For more information, refer to the *Security Sub Model* section.

For more information about sequences or stored procedures and your database, refer to *Loading Policy and Sequences*.

Loading Cost Center Dimensions

The cost center dimensions do not have to be pre-loaded.

The TAM CBU fact tables do not depend on the information in the cost center dimensions. This makes on-demand loading irrelevant.

Tables:

- NMY_COST_CENTER_DIM
- NMY_CC_CONTRACT_LNK
- NMY_COST_CENTER_LNK

The CID2CBU loader synchronizes the cost center dimensions with the CID.

Fact Table Loading Policy

Tables:

- NMY_MAIN_INVOICE_FACT
- NMY_CONTRACT_INVOICE_FACT
- NMY_BUDR_FACT

Due to the volume of these tables, you should load these tables by using batches. The batch must cope with on-demand loading for dimensions that implement it.

You must load NMY_MAIN_INVOICE_FACT first. You can then load NMY_CONTRACT_INVOICE_FACT and NMY_BUDR_FACT. The reconciliation is done either on BILL_ACCT + BILL_PERIOD or on MAIN_INVOICE_UIDN if any.

When loading these fact tables:

- If customer data dimension are pre-loaded, new facts that do not match must be discarded.
- If customer data dimension are on-demand, new facts must create any missing entries.

Loading Guidelines:

1. Determine the reconciliation algorithm for each reference dimension keys. Some keys may be degenerated to a single value if it is not used.
2. Determine the reconciliation algorithm for conforming dimension keys. You can use the business key or the UIDN value.
3. Determine the fact attributes that are required by your reports. Unused fact attributes can be ignored, as they are not mandatory.
4. Implement a loading order mechanism taking into account that Contract Invoice and BUDR facts are independent but require Main Invoice facts.
5. In pre-loaded scenario:
    - Load facts from the fact stream.
    - Discard facts when customer data dimension entries are not found.
6. In on-demand scenario:
    - Load facts from the fact stream.
    - Create new customer data dimension entries when necessary.
    - Update newly created dimension entries. This requires populating missing business keys and attributes.

Index