

---

# HP Service Quality Management Solution



## **SQM Service Designer V3.0 User Guide**

**Edition: 2.0**

**for Microsoft Windows Operating System**

**Nov 2011**

© Copyright 2011 Hewlett-Packard Company, L.P.

## Legal Notices

### Warranty

The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

### License requirement and U.S. Government legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

### Copyright notices

© Copyright 2011 Hewlett-Packard Development Company, L.P.

### Trademark notices

Java™ and all Java based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

The information provided by TM Forum is the property of TM Forum or its licensors and is protected by copyright and other intellectual property laws of the United States, Canada and other jurisdictions.

Oracle® is a registered U.S. trademark of Oracle Corporation, Redwood City, California.

Windows®, and Windows NT® are U.S. registered trademarks of Microsoft Corporation.

# Contents

<b>Chapter 1 .....</b>	<b>8</b>
1.1 Service Designer Components .....	8
1.1.1 Telco Palette .....	8
1.1.2 Transducer .....	9
1.1.3 Reverse-Transducer .....	9
1.2 Service Designer Workflows .....	9
1.2.1 Reverse-Transducer Workflow .....	9
1.2.2 Transducer Workflow .....	11
<b>Chapter 2 .....</b>	<b>12</b>
2.1 Using the Telco Palette .....	12
2.1.1 Enabling the Telecom Universe Profile .....	13
2.2 Importing SMF UML workspace .....	16
2.3 Creating a Service Model .....	17
2.3.1 Defining a CI Type .....	17
2.3.2 Defining a CIT Attribute .....	20
2.3.3 Defining a Relationship .....	21
2.3.4 Defining an inheritance Relationship .....	22
2.3.5 Defining a Calculated Link triplet .....	23
2.3.6 Creating a link for Existing CIT .....	23
2.4 Creating a Content Pack .....	25
2.4.1 Defining a Business Rule .....	26
2.4.2 Defining an Health Indicator .....	29
2.4.3 Defining a Key Performance Indicator .....	31
2.4.4 Defining an HI Assignment .....	33
<b>Chapter 3 .....</b>	<b>36</b>
3.1 Exporting a UML Model .....	36
3.2 Translating a UML Model .....	38
3.3 Deploying a Service Model in BSM .....	39
3.3.1 Deploying CITs and Relationships in BSM .....	39
3.3.2 Deploying BRs, HIs and KPIs in BSM .....	41
<b>Chapter 4 .....</b>	<b>43</b>
4.1 Retrieving a Service Model .....	43
4.1.1 Exporting CITs and Relationships from BSM .....	43
4.1.2 Exporting Content Pack from BSM .....	44

4.2	Translating a Service Model .....	44
4.3	Importing and Viewing a Service Model in UML .....	46
<b>Chapter 5</b>	<b>.....</b>	<b>48</b>
5.1	Configuring Error Tracing.....	48
5.2	Configuring Log Generation.....	50
5.3	Managing Licenses .....	51
<b>Appendix A</b>	<b>.....</b>	<b>53</b>
<b>Index</b>	<b>.....</b>	<b>58</b>

# Preface

This guide is designed to be used as a manual of operations for the HP SQM Solution Service Designer tool that is used to design and modify service models in UML, and translate service models from UML to BSM-recognized models and vice versa.

This document also presents and explains the various concepts regarding different components that are included in the Service Designer tool.

The guide is organized in a manner that the user is initially given an overview of the Service Designer tool and introduction of each major component integrated in the tool. The guide then uses the high-level overview to drill down and explain each component and the various stages of the Service Designer workflow using real-life examples.

## Intended Audience

This document is intended for the following users:

- SQM Solution user
- SQM Solution administrators.

## Abbreviations and Acronyms

The following table describes the abbreviations and acronyms used in this document.

Abbreviation	Description
<b>BSM</b>	Business Service Management
<b>CI</b>	Configuration Item
<b>CIT</b>	Configuration Item Type
<b>DDM</b>	Discovery & Dependency Mapping
<b>DDP</b>	Discovery & Dataload Pack
<b>DSA</b>	Data Source Adapter
<b>HI</b>	Health Indicator
<b>KPI</b>	Key Performance Indicator
<b>MA</b>	Monitoring Adapter
<b>OSS</b>	Operations Support System
<b>RTSM</b>	Run-Time Service Model
<b>SA</b>	Service Adapter
<b>SID</b>	Shared Information Data
<b>SiS</b>	SiteScope

<b>SMF</b>	Service Management Foundation
<b>SQM</b>	Service Quality Management
<b>TMF</b>	TeleManagement Forum
<b>uCMDB</b>	Universal Configuration Management Data Base

## Associated documents

The HP Business Availability Center and SiteScope documents are available at: <http://support.openview.hp.com/selfsolve/manuals>




Additional SQM Solution materials (like the SQM Solution product briefs) and information about SQM Solution updates are available at: <http://www.hp.com/cms>

## Typographic Conventions

This document uses the following conventions to identify special information:

Convention	Information Type/Example
[ ] (square brackets)	Interface components requiring user actions e.g. Buttons. Ex: Click [Finish] to complete the Import wizard.
( ) [round brackets]	Supplementary information <i>Ex</i> : Configuration Item (CI).
<b>Bold type</b>	Fields names, menus, window pane names Ex of menus: <b>Admin</b> → <b>Service Level Management</b> → <b>Repository</b> .
<i>Italic type</i>	Important information and/or concepts. Ex: The output is an <i>.XMI</i> file.

## Symbols Used in this Guide

Symbols	Information
	<b>Note</b> Draws your attention to additional information about a software function/feature.
	<b>Important</b> Draws your attention to important information regarding the proper usage of a software function/feature.
	<b>Caution</b> Draws your attention to an important warning.

## Support

Please visit our HP Software Support Online Web site at:

<http://support.openview.hp.com/support.jsp>

HP Software online software support provides customer self-solving capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support site to:

- Search for knowledge documents of interest
- Submit enhancement requests online
- Download software patches
- Submit and track progress on support cases
- Manage a support contract
- Look up HP support contacts
- Review information about available services
- Enter discussions with other software customers
- Research and register for software training.

# Chapter 1

## Service Designer Overview

The SQM Service Designer solution is designed to aid the user in creating, manipulating and deploying service models with utmost ease and precision. Service Designer employs two main processes, the first process involves designing the service model using a conventional third party UML modelling tool (*Borland Together 2008 Release 3*), and then converting and deploying the model in BSM. The second process involves manipulating existing BSM service models by importing a model in the UML modelling tool and then modifying it. These processes and the various components of the SQM Service Designer are discussed in more detail in the following sections.

### 1.1 Service Designer Components

The Service Designer Solution is composed of various independent components which are used in different stages of the service design/modification process. In the first version of the Service Designer, these components support the following model information:

- Definition of CI types and relationships
- Relationships between CI types
- Definition of Health Indicators (HI)
- Definition of Key Performance Indicators (KPI)
- Definition of Business Rules (BR)








The Service Designer components are discussed below in further detail.










SQM Service Designer supports creation of HIs, KPIs and Business Rules. The delivered HIs, KPIs and BRs by BSM and SMF 3.0 are included in a UML workspace.

#### 1.1.1 Telco Palette

A special customized Telco Palette is integrated in the UML modelling tool (*Borland Together 2008*) and appears as part of the standard UML design Palette. The Telco Palette contains several UML stereotypes specific to SQM Solution modelling. The Telco Palette is a user-friendly feature that allows the user to define a service model using the following stereotypes recognized in BSM:

- CI Type (CIT , CIT\_ATTRIBUTE , ATTRIBUTE\_OVERRIDE )
- CMDB\_RELATION, CALCULATED\_LINKS\_TRIPLET , Generalization 
- Business Rule (BR\_DASH , BR\_SLM )



- Health Indicators (HI\_DASH , HI\_SLM )
- KPI (KPI\_DASH , KPI\_SLM )
- HI Assignment (HI\_CONFIGURATION , HI\_ASSIGNMENT\_DASH ,  
HI\_ASSIGNMENT\_SLM )



For more information about using the Telco Palette, refer to *Chapter 2 Designing Service Models*.

### 1.1.2 Transducer

Transducer is a command-line tool which is used to convert a UML model file to a BSM/RTSM model file, which can then be imported and deployed in RTSM UI of BSM, as required. The input for the Transducer process is an “.XMI” file exported by the UML modelling tool (*Borland Together 2008 Release 3*), and the output is a “.ZIP” file containing CIT definitions which may be imported using RTSM UI of BSM and a “.XML” file containing HI, KPI, BR and Propagation Rule definitions which may be imported with Content Pack UI of BSM.



For more information about working with the Transducer tool, refer to *Chapter 3 Deploying Service Models in BSM*.

### 1.1.3 Reverse-Transducer

Reverse-Transducer is also a command-line tool similar to the *Transducer*. The main function of this tool is to convert a model file and Content Pack file to a UML model file, which can then be imported and deployed in the UML modelling tool (*Borland Together 2008 Release 3*). The input for the *Reverse-Transducer* process is a set of model (”.ZIP”) files, containing definitions retrieved from BSM/RTSM, and a set of Content Pack (”.XML”) files. Its output is an .XMI file which may simply be imported in the UML modelling tool.



For more information about working with the *Reverse-Transducer* tool, refer to *Chapter 4 Importing Service Models in UML*.

## 1.2 Service Designer Workflows

As we have discussed earlier that the Service Designer essentially employs two processes both with different inputs and outputs, this section presents a depiction for the workings of these processes.

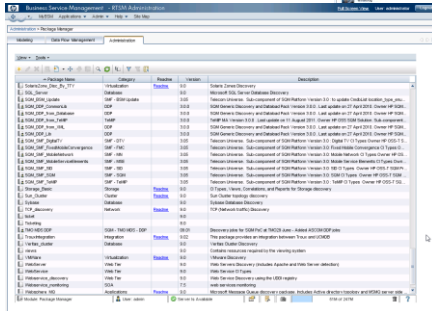
### 1.2.1 Reverse-Transducer Workflow

The Reverse-Transducer workflow involves retrieving and manipulating existing BSM service models by converting and then importing a model in the UML modelling tool (*Borland Together 2008 Release 3*).

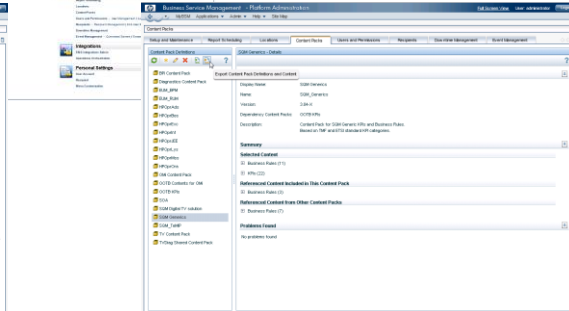
## BSM Admin UI



## RTSM Admin UI

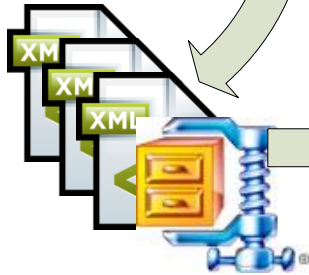


## Content Pack Admin UI



SELECT uCMDB Package  
EXPORT Package

SELECT Content Pack  
EXPORT Content Pack

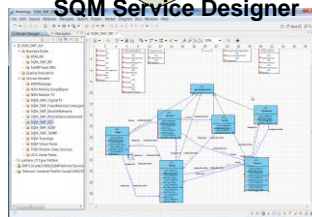


ReverseTransduce



IMPORT  
.. MODELING  
... XMI FILE

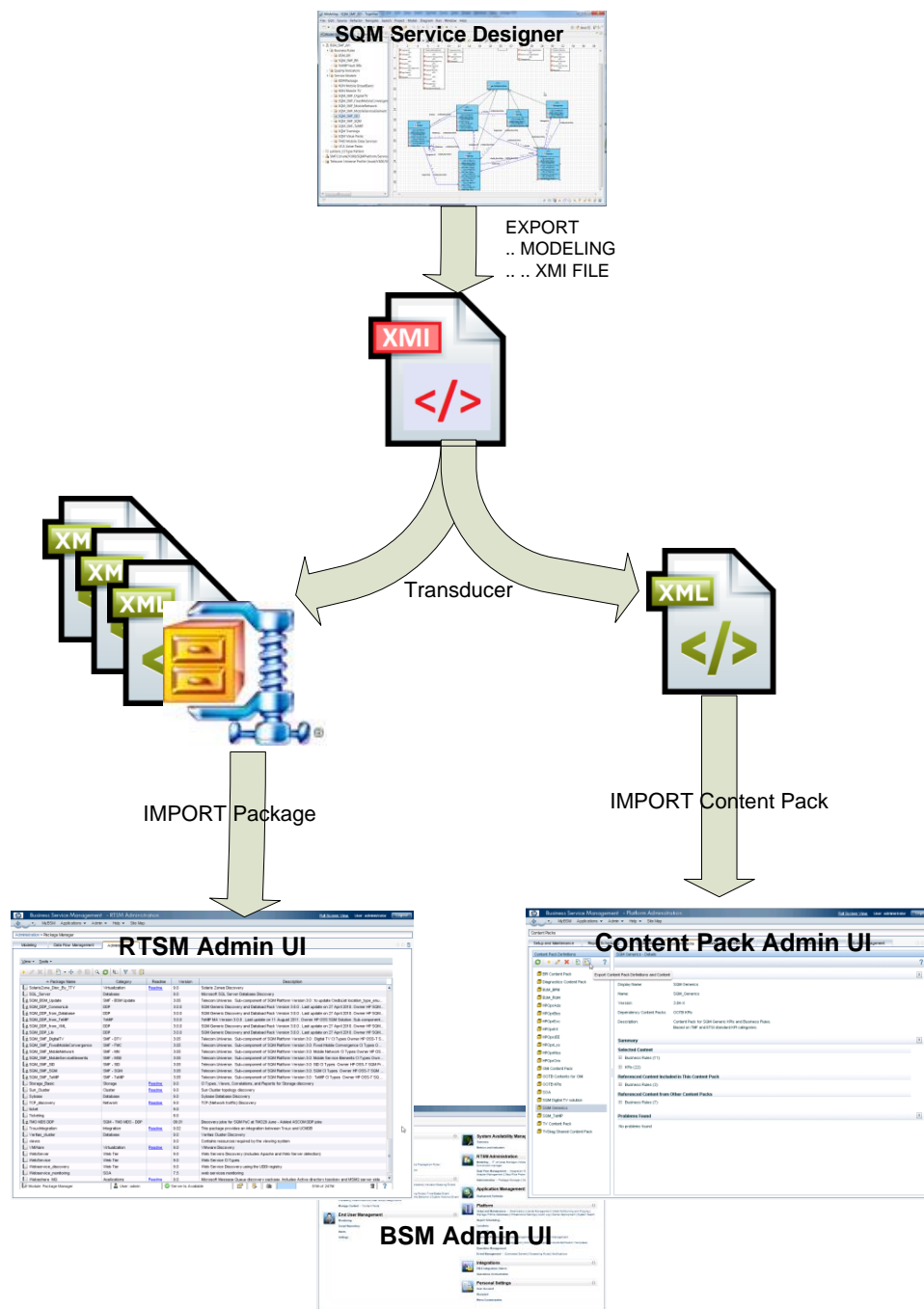
## SQM Service Designer



For more information about the Reverse-Transducer workflow, refer to *Chapter 4 Importing Service Models in UML*.

## 1.2.2 Transducer Workflow

The Transducer workflow involves designing a new service model or modifying an existing model imported from BSM, using a conventional third party UML modelling tool (*Borland Together 2008*), and then converting and deploying the model in BSM/RTSM and HI, KPI, BR in BSM/Content Pack.

















For more information about the Transducer workflow, refer to *Chapter 3 Deploying Service Models in BSM*.



# Designing Service Models

This chapter contains the definition and usage of the Telco Service Designer Palette and the procedure of creating a service model in *Borland Together* using the several available stereotypes specific to Telco SQM Solution modelling.

## 2.1 Using the Telco Palette

The Telco Service Designer Palette is integrated to the UML modelling tool (*Borland Together 2008 Release 3*), and is designed to provide accessibility to the user for creating service models that are recognized in BSM. The following figure shows the stereotypes and their associated icons available in the Telco Palette.


Stereotype	Icon	Description
CIT		UML class that represents a CI Type
CIT_ATTRIBUTE		UML class that represents a CI Type Attribute
ATTRIBUTE_OVERRIDE		UML class that represents a CI Type Attribute Override
CMDB_RELATION		UML class that represents a Relationship between 2 CI Types.
CALCULATED_LINKS_TRIPLET		UML class that represents a Calculated Link triplet between 2 CI Types
Generalization		UML inheritance link that creates extend value
BR_DASH		UML class that represents a BR dedicated for BSM Service Health
BR_SLM		UML class that represents a BR dedicated for BSM SLM
BR_TOOLTIP_PARAMETER		UML class that represents a BR tooltip parameter
HI_DASH		UML class that represents a HI dedicated for BSM Service Health
HI_SLM		UML class that represents a HI dedicated for BSM SLM
HI_ASSIGNMENT_DASH		UML class that represents a HI Assignment dedicated for BSM Service Health
HI_ASSIGNMENT_SLM		UML class that represents a HI Assignment dedicated for BSM SLM
HI_CONFIGURATION		UML class that represents a HI configuration

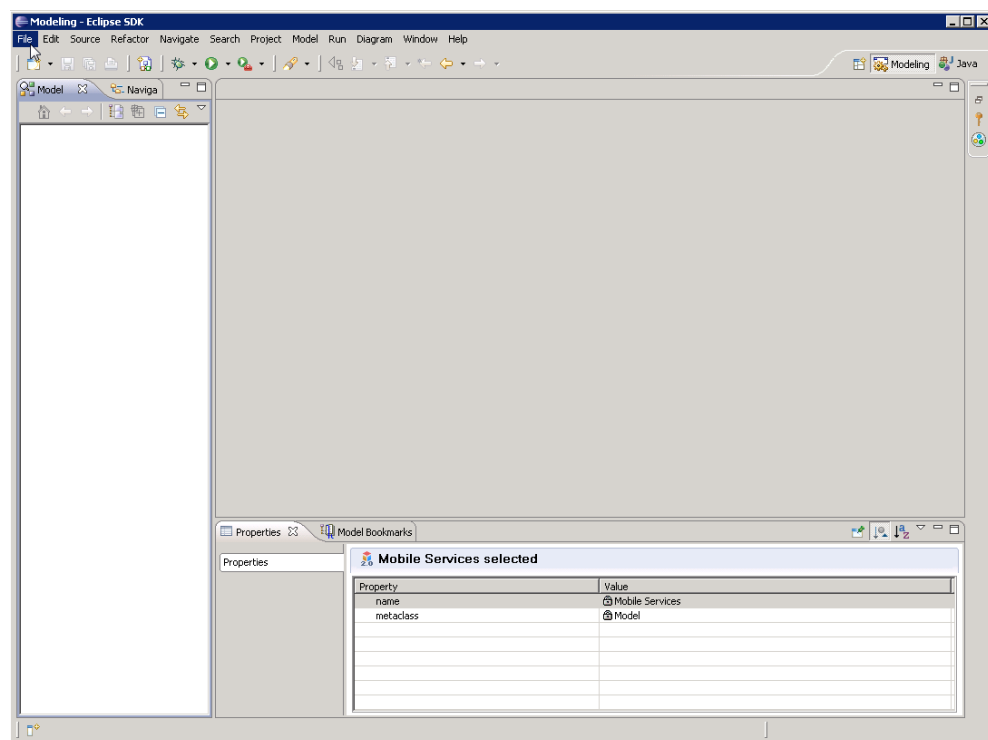
KPI_DASH		UML class that represents a KPI dedicated for BSM Service Health
KPI_SLM		UML class that represents a KPI dedicated for BSM SLM

The following sections discuss the usage of the Telco Palette and what is required to start working with the palette.

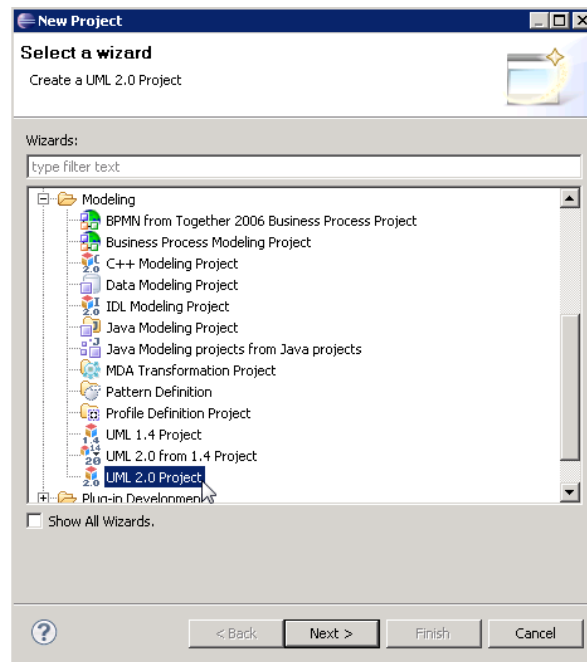
### 2.1.1 Enabling the Telecom Universe Profile

In order for the Telco Palette to display in *Together*, the *Telecom Universe Profile* plug-in must be installed and enabled in your *Together* UML project. Once you install the *Borland Together 2008* application that is provided with the *Service Designer Installation Kit*, the *Telecom Universe Profile* plug-in will be installed automatically. Once installed, you will be required to enable the profile in order to view the Telco Palette in your UML modelling project. To enable the Telecom Universe profile, you will be required to perform the following steps:

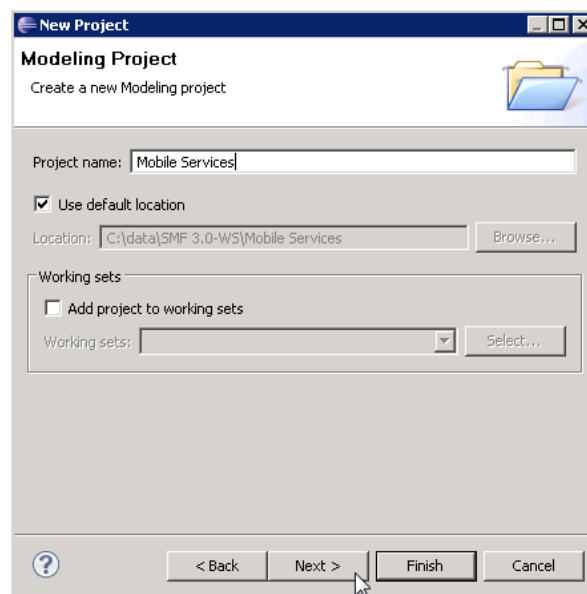
1. From the **Start** menu browse and select the program option to launch *Borland Together 2008*. The Modelling-Together window displays.
2. Make sure that the *Together Perspective* is set to **Data Modelling**. You can set *Perspective* by clicking  [Open Perspective] on the top-right corner of the *Together* main window, and then selecting **Data Modelling** from the pop-up menu, as shown in the figure below:



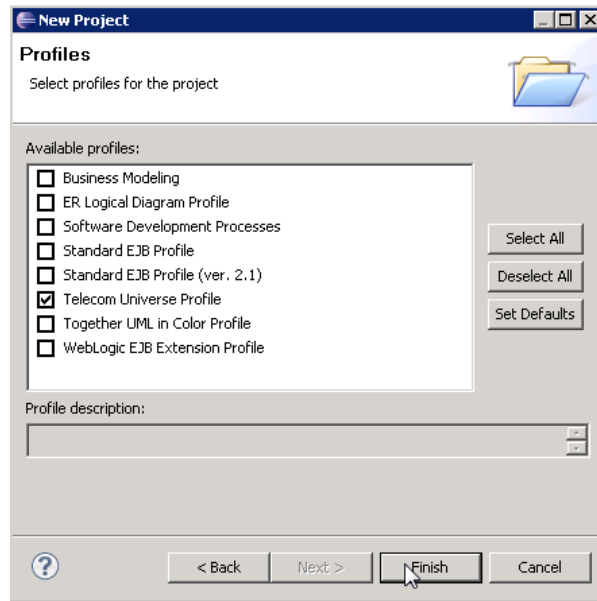
3. Select **File → New → Project**. The New Project – Select a wizard dialog displays.



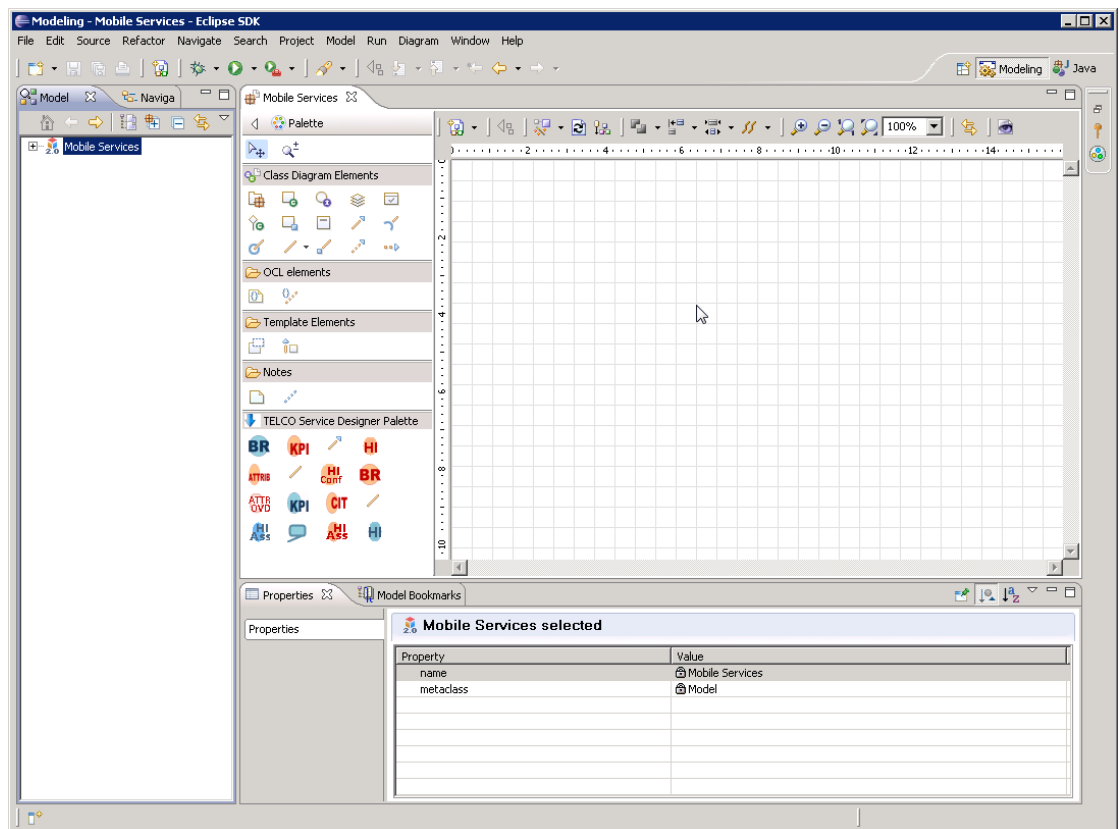
4. Select **Modelling** → **UML 2.0 Project** from the Wizards list and the click [Next]. The New Project – Modelling Project dialog displays.



5. Enter the **Project name** and then click [Next]. The New Project – Modelling Settings dialog displays.
6. Click [Next]. The New Project – Profiles dialog displays. Select *Telecom Universe Profile* from the Available profiles list.



7. Click [Finish] to create a new UML modelling project. The newly created UML project workspace displays.



8. The *Telco Service Designer Palette* is available in the **Palette** pane of *Borland Together*, as shown in the above figure.

## 2.2 Importing SMF UML workspace

In order to facilitate a new Service Model based on current SMF 3.0, an UML workspace is delivered.

- I. Create a folder for your workspace
- II. Start SQM SD using this new workspace
- III. Create a new UML 2.0 project like “MyModel”:
  - From the Menu, Select “**File**” > “**New**” > “**Project**”
  - On the “**New Project**” Wizard, locate “**Modelling**” folder, and expand it
  - Select UML 2.0 project (see screenshot below), and click <**Next**>
  - Enter a project name, and click <**Finish**>

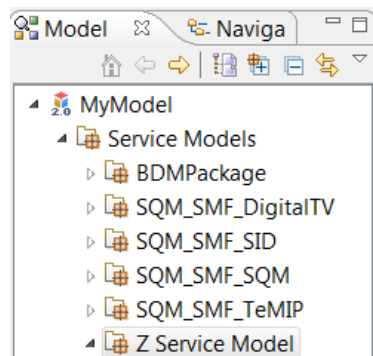


Do not use space (“ ”) character in your project name.

- IV. Use **Import -> General -> File System**
  - Select your project, Right Click
  - Select “**Import**”, in **Import dialog** select **General** and **then File System**
  - Click <**Next**>, enter full path of predefined UML project

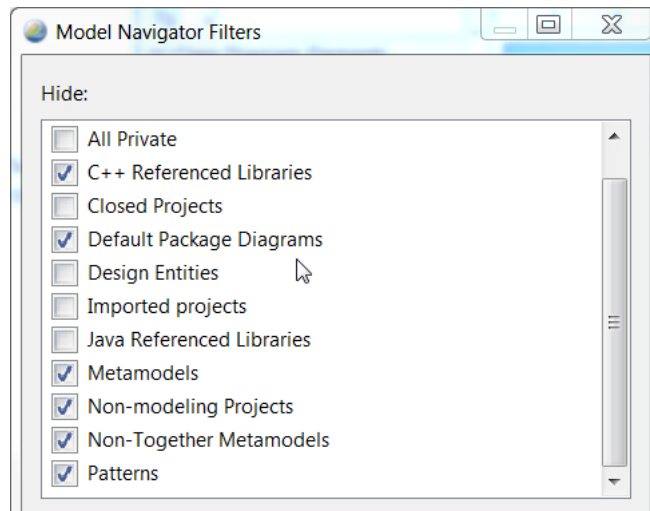
Default path is “<SD\_Install\_Dir>\SD\_Project\SMF3”, like “C:\Program Files\HP\SQM\Service\_Designer\SD\_Project\SMF3”.

- Click [**Select All**] and then click [**Finish**]



- V. **Recommendation:** Set the Borland Together Filter to the following:





## 2.3 Creating a Service Model

The Telco Service Designer Palette allows you to create a UML model in *Together* that can be converted and imported in BSM/RTSM. The following components of a service model can be defined in *Together*:

- CI Type
- CIT Attribute & CIT Attribute Override
- CMDB Relationships & Calculated Triplet Links

All the above components are represented as UML stereotypes in the Telco Palette.



Supported Service Models are mapped to BSM Models(refer to *Appendix A*). More information could be found in the current BSM.

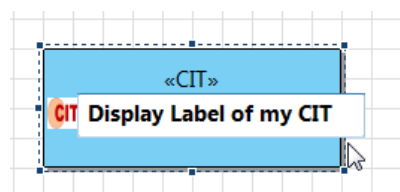
Any Service Model is contained in a specific package while exported from BSM. It is a pre-defined requirement for creating Model that service model(s) be contained in its package. This package name will be used later as Service Designer command line tool's parameter.

More information about how to user Service Designer Command Tool, refer to [Translating a UML Model](#).

### 2.3.1 Defining a CI Type

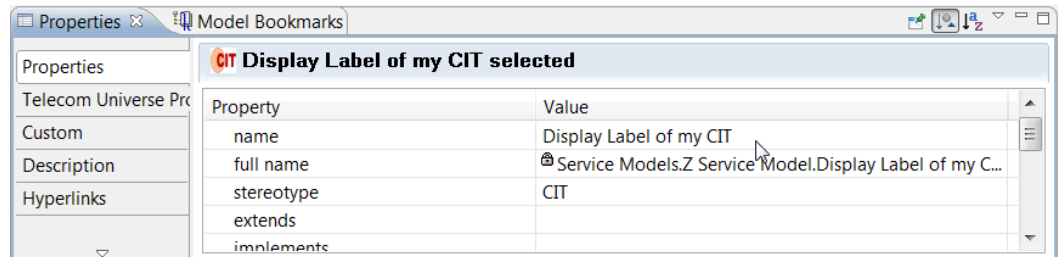
To define a CI Type as a UML class in *Together* using the Telco Palette, you will be required to perform the following steps. The following description contains information only about the required properties when defining a CIT object:

1. On your working package, from the **Telco Service Designer Palette**, select the **CIT** stereotype and then draw the object in the empty workspace
2. Enter the Name of your newly created CIT:

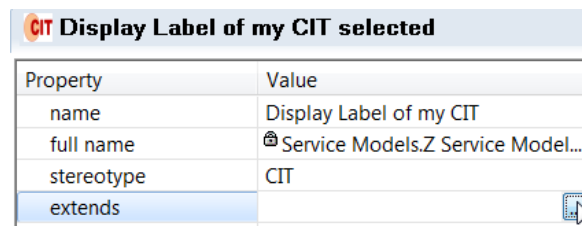


**Note:** Length of CIT “Name” (“Display Label” in BSM) is limited to 50 characters.

3. Select the newly created CI Type object. The **Properties** pane displays at the bottom of the main application window. And verify the “name” field:

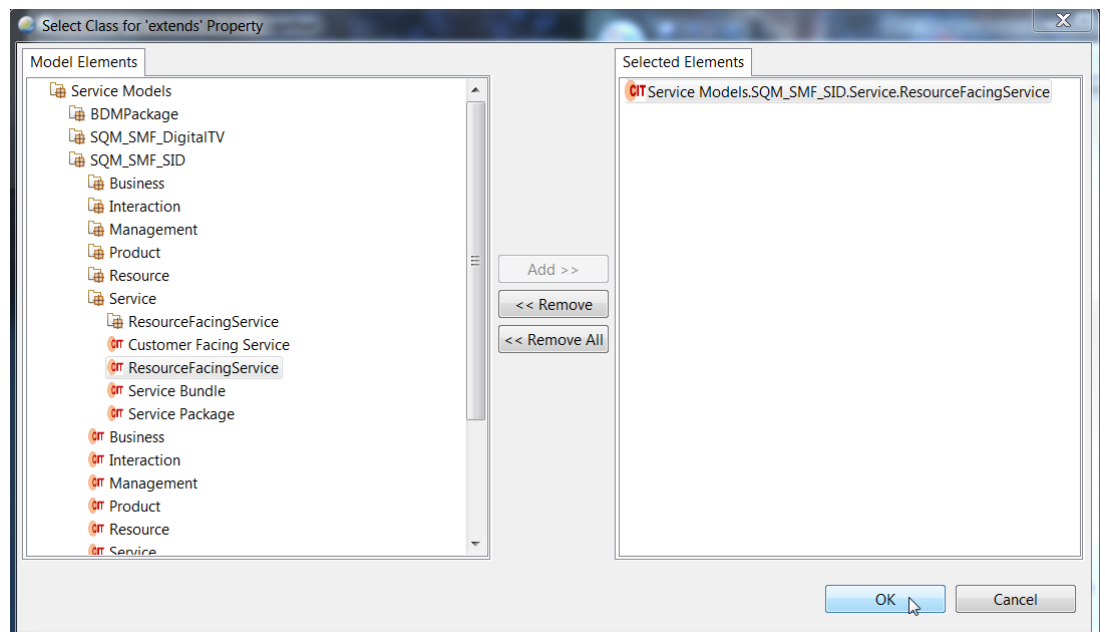


4. To set the extends (define parent CIT from which your newly created CIT inherit attributes and properties), click on right side of value field

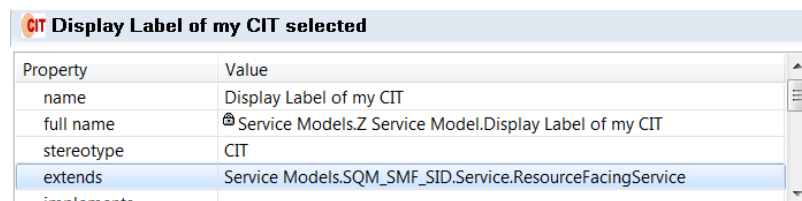


**Note:** This is a standard way to set CIT parent, another way refer to section [Defining an inheritance Relationship](#).

5. On the pop-up window, you will select its parent CIT, click < Add>> >, and <OK>

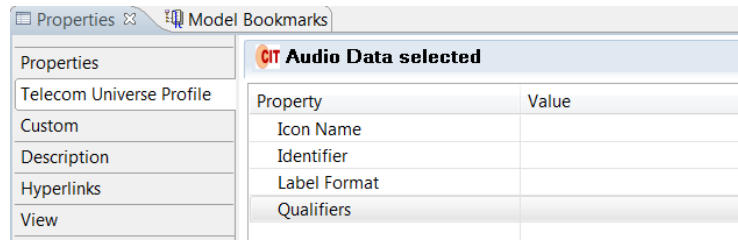


6. To obtain this in “extends” field:



**Note:** make sure you do have only one CIT as value.

7. From the **Properties** pane, select **Telecom Universe Profile** tab, as shown in the figure below.



8. There are four properties, but only one is mandatory: Identifier
  - **Mandatory:** Enter an **Identifier** for the new CI Type. This CIT property is required in BSM.

**Note:** Length of CIT “**Identifier**” (“**Name**” in BSM) is limited to 30 characters, and only use character a-z, A-Z, 0-9 and “\_” “..”

- **Optional:** Enter an **Icon Name** for the new CI Type.

**Note:** if you do not set an icon, it will inherit from the parent CIT.

- **Optional:** Enter a **Label Format** for the new CI Type.

You build a label by selecting attributes in the CI Type Attributes pane and adding them to the Format pane, using the operators in the Format pane to connect them.

The label appears as the title under a CI of the new CI Type. The label definition can be customized to include different attribute values. For example, if the function label of the node CIT is composed of hostname and network, the displayed label is:  
server1 10.0.65.0.

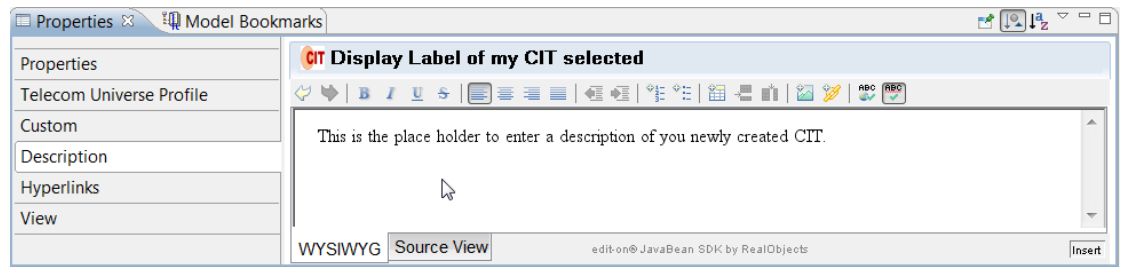
Labels can also be created using regular expressions.

**Note:** check BSM documentation about “Qualifier Tab” to get more [Information \(System Administration > Modelling > Modelling > CI Type Manager > Create Configuration Item Type/Relationship/Calculated Relationship Wizard > Default Label Page\)](#).

- **Optional:** Enter a **Qualifier** for the new CI Type.

**Note:** check BSM documentation about “Qualifier Tab” to get more [information \(System Administration\) > Modelling > Introduction to RTSM > Topology Query Language > Query Node/Relationship Properties Dialog Box\)](#).

9. Finally, you could enter a description:

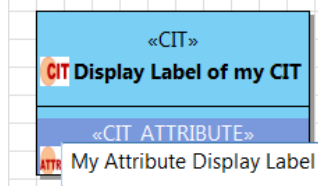


**Note:** Length of CIT “Description” is limited to 500 characters.

## 2.3.2 Defining a CIT Attribute

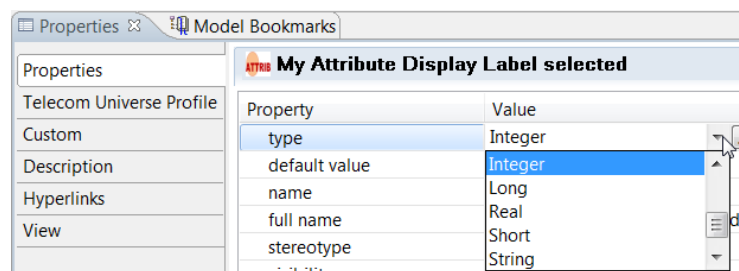
To define attributes of an existing CI type, in *Together*, using the Telco Palette, you will be required to perform the following steps. The following description contains information only about the required properties when defining CIT attributes:

1. On your working package, from the **Telco Service Designer Palette**, select the **CIT\_ATTRIBUTE** stereotype and then click inside the CIT object that already exists in the workspace.
2. Enter the Name of your newly created **CIT\_ATTRIBUTE**:



**Note:** Length of CIT\_Attribute “Name” (“Display Label” in BSM) is limited to 100 characters.

3. In the **Properties** Pane, set different properties:
  - i. **Mandatory:** select the data **type**:



- ii. **Optional:** set a default Value
4. In the **Telecom Universe Profile** tab, set other properties for the CIT attribute:

Properties	
Telecom Universe Profile	
Custom	
Description	
Hyperlinks	
View	

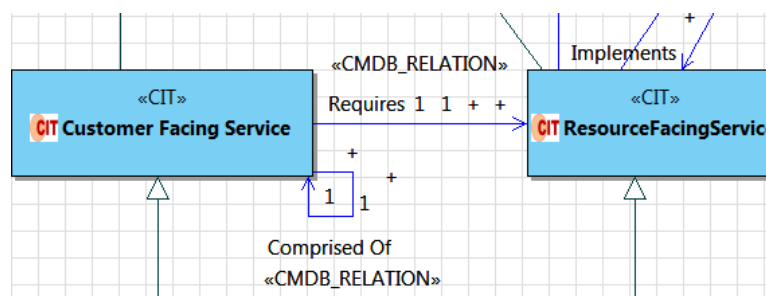
My Attribute Display Label selected	
Property	Value
Asset Data	<input type="checkbox"/> false
Change Monitored	<input type="checkbox"/> false
Comparable	<input type="checkbox"/> false
Enable Auto Truncate by DDM	<input type="checkbox"/> false
Identifier	Myidentifier
Index	<input type="checkbox"/> false
Key	<input type="checkbox"/> false
Lower Case	<input type="checkbox"/> false
Password	<input type="checkbox"/> false
Required	<input type="checkbox"/> false
Size	
Upper Case	<input type="checkbox"/> false
Visible	<input type="checkbox"/> false

- i. **Mandatory:** enter a new unique **identifier**:  
*Note:* Length of CIT\_Attribute “**Identifier**” (“**Name**” in BSM) is limited to 100 characters, and do not use “ ” (space) character.
  - ii. **Optional:** set the attribute size in case of String.
  - iii. **Optional:** change some Boolean attribute if needed:
5. In the **Description** tab, you could enter a description for this CIT attribute.  
*Note:* Length of CIT “**Description**” is limited to 100 characters.

### 2.3.3 Defining a Relationship

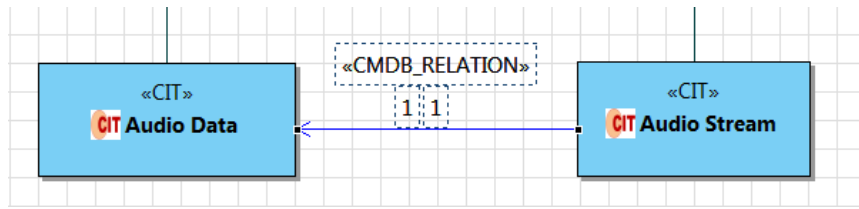
A relationship defines the link between two CIs. Relationships represent the dependencies and connections between the entities in your IT/TELCO environment. On the BSM RTSM Administration UI, When you select relationships or calculated relationships in drop-down box in the CI Types pane, the topology map displays all the valid instances of CITs linked by the selected relationship.

In UML model, a relationship is represented as “Kernel Association” (“**CMDB\_RELATION**”):

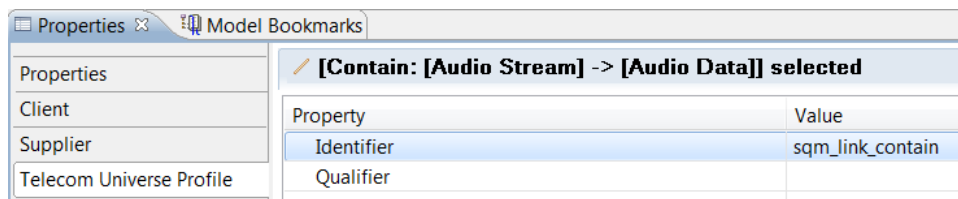


To define a relationship between two objects in a UML model using the Telco Palette, you will be required to perform the following steps:

1. From the Telco Service Designer Palette, select the **CMDB\_RELATION** stereotype.
2. Draw the relationship from the child object to the parent object, such as shown in the figure below.




3. Select the newly created relationship. The **Properties** pane displays at the bottom of the main application window.
4. Enter a **label** name for the new relationship, which will be converted to the display name of the CI relationship in BSM.
5. From the **Properties** pane, select **Telecom Universe Profile** tab, as shown in the figure below.

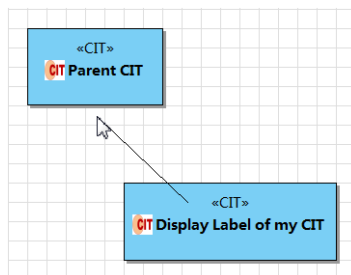


6. Enter an **Identifier** for the relationship. This property is required in BSM. Please ensure that the identifier matches the corresponding BSM relationship name (case-sensitive, no space (" "), and no dash ("-")).

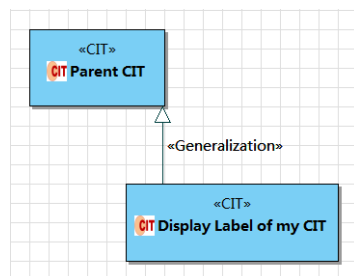
### 2.3.4 Defining an inheritance Relationship

You define an inheritance relationship when you want a CIT that inherit the attributes and behaviour (such as the label function) of the parent CIT. There is already one way to create *inheritance* relationship in section [Defining a CI Type](#). Here is another way.

1. Select  from the **Class Diagram Elements** palette.
2. Click the child CIT, there will be a line attached to mouse cursor. Move mouse to parent CIT and click again.



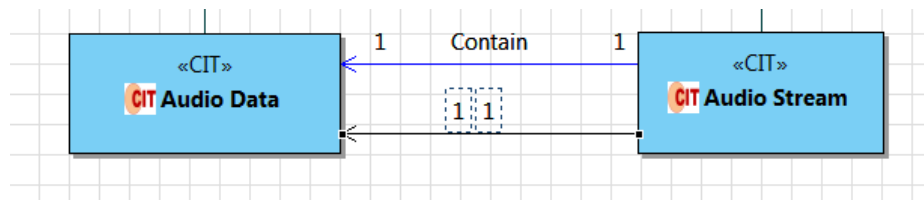
3. The inheritance relationship is created between the two CI types, as shown in the figure below.



## 2.3.5 Defining a Calculated Link triplet

SMF 3.0 is providing RTSM relationship and Calculated Links Triplet for the SID domains, so any TELCO CIT will inherit from them. But you might need to have KPI propagation between IT and TELCO CIT's; in this case, you will have to create a new Calculated Links Triplet by performing the following steps:

1. From the Telco Service Designer Palette, select the **CALCULATED\_LINK\_TRIPLET** stereotype.
2. Draw the relationship from the child object to the parent object, such as shown in the figure below.



3. Select the newly created relationship. The **Properties** pane displays at the bottom of the main application window.
4. Enter a **label** name for the new relationship, which will be converted to the display name of the CI relationship in BSM, and modify the **name**.
5. From the **Properties** pane, select **Telecom Universe Profile** tab, as shown in the figure below

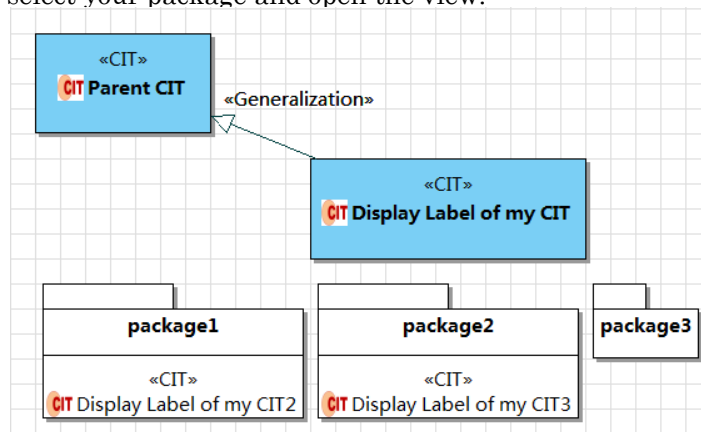
[Kernel Association: [Audio Stream] -> [Audio Data]]	
Property	Value
CalculatedLinkType	impact_dependency
Identifier	sqm_link_contain
IsForward	<input checked="" type="checkbox"/> true

6. Enter an **Identifier** of the relationship. This property is required in BSM. Please ensure that the identifier matches the corresponding BSM relationship name (case-sensitive, no space (" "), and no dash ("-")).
7. Enter the **CalculatedLinkTriplet** (value should be "impact\_dependency")
8. Check "**isForward**" if the relationship direction is from "**source**" to "**target**" (default is false: it means the impact is from "**target**" to "**source**"); uncheck **false** to have set it to **true**.

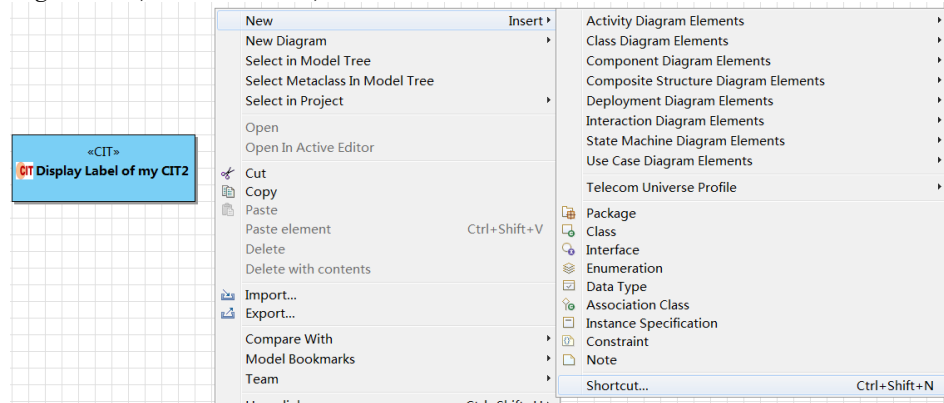
## 2.3.6 Creating a link for Existing CIT

It is possible there are several packages, and in each package there are several CITs inheritance from the same CIT, you should not create the parent CIT in all the packages. In this situation, you will need to create CIT link by performing the following steps:

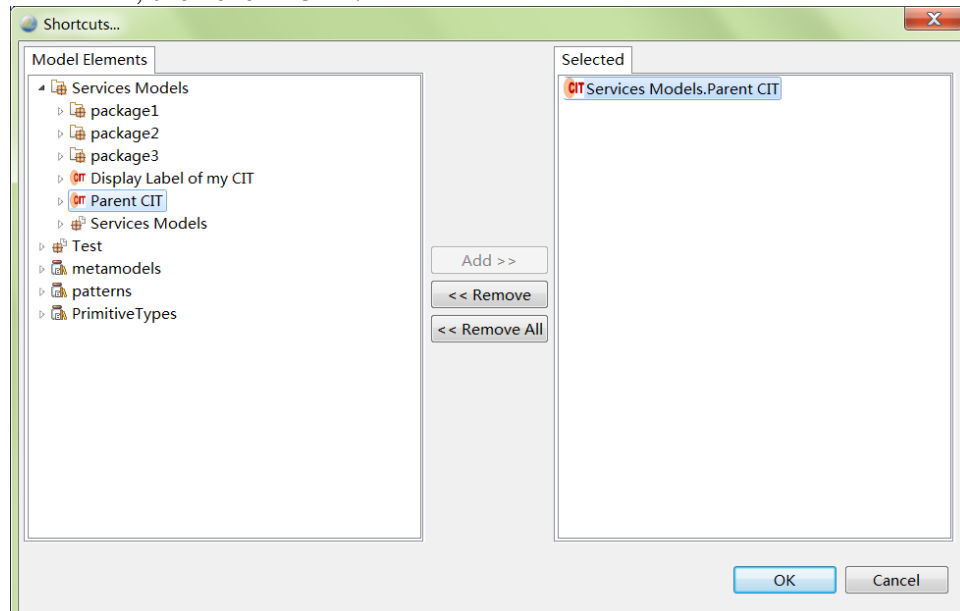
1. Suppose there are existing packages as shown below. From the view pane, select your package and open the view.



2. Right click, select “New”, and select “Shortcut”.

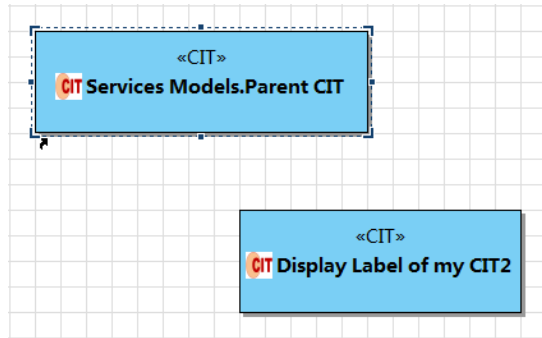


3. After “Shortcuts” windows shows, select parent class on the left and click <ADD>>, then click <OK>.

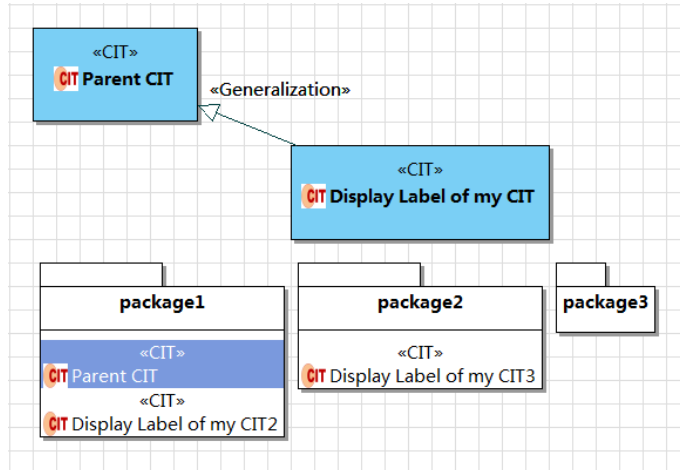


4. A CIT Link shows in the view as below. Then you can add inheritance between the two CITs.





5. Here is an overview of the view.



Link and real CIT models are referenced to each other, so you can delete the link but you cannot delete real CIT model.

Link cannot be copy/cloned due to modeling tool functionality. If you need multiply links, you need to perform all the steps in each package.

## 2.4 Creating a Content Pack

The Telco Service Designer Palette allows you to create a UML model in *Together* that can be converted and imported in BSM/Service Health and BSM/SLM repositories. The following components of a service model can be defined in *Together*:

- Business Rules
- Health Indicators
- KPIs



Transducer checks the references between BR, HI, KPI and so on. So if using existing BSM Rule, it is strongly recommended that import all BSM, SQM BR before creating any HI/KPI.

All the above components are represented as UML stereotypes in the Telco Palette.

## 2.4.1 Defining a Business Rule

To define Business Rule in a UML model using the Telco Palette, you will be required to perform the following steps:

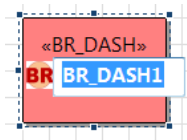
1. From the **Telco Service Designer Palette**, select the **BR\_DASH** or **BR\_SLM** stereotype. Your choice should depend on whether you want to define a *Dashboard Business Rule* or *SLM Business Rule*.
2. Draw the object into a specific package of the workspace.



Any Service Model is contained in a specific package while exported from BSM. It is a pre-defined requirement for creating Model that service model(s) be contained in its package. This package name will be used later as Service Designer command line tool's parameter.

More information about how to use Service Designer Command Tool, refer to [Translating a UML Model](#).

3. Select the newly created Business Rule object. The **Properties** pane displays at the bottom of the main application window.



4. Enter a name of the Business Rule in the **name** text field. The name will be mapped in BSM.

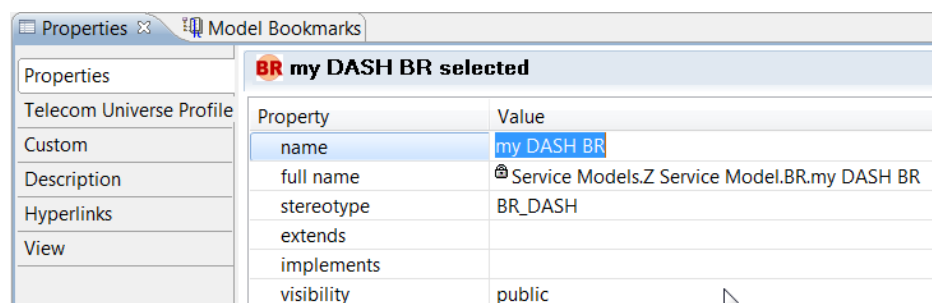


It is recommended to follow this naming convention to have unique name within BSM:

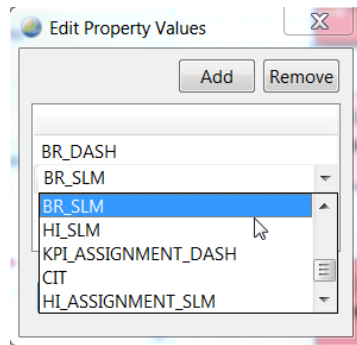
BR-<Name of the Business Rule>-(Service Health|SLM)

**Example:** BR-My Rule(Service Health)

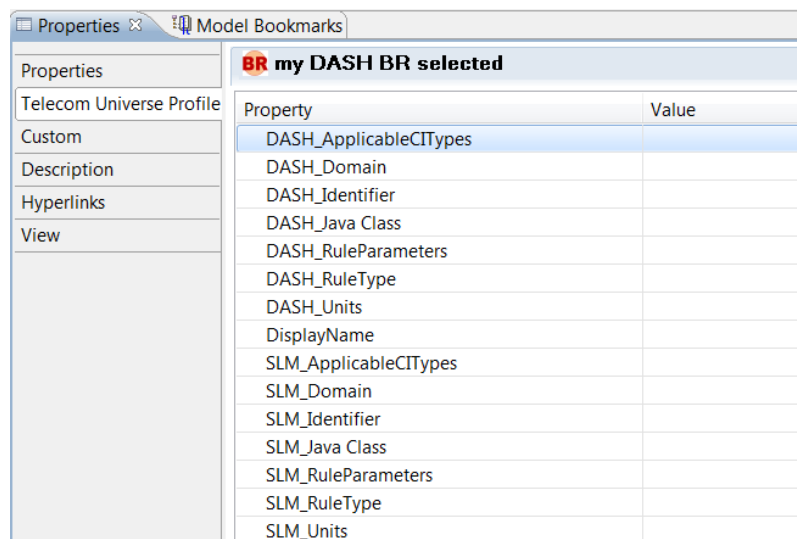
5. Select the newly created BR object. The **Properties** pane displays at the bottom of the main application window, and Verify the **name**:



6. If the scope of BR is “**Service Health**” and “**SLM**”, then select “**stereotype**” value, and on the right side click
7. On the pop-up window, click <Add>
8. Use the drop down menu, to select “**BR\_SLM**”, and click <OK>



6. From the **Properties** pane, select **Telecom Universe Profile** tab, as shown in the figure below.



7. Mandatory parameters:

- (1) **DisplayName:** The name will be mapped in BSM display label.
- (2) **DASH\_ApplicableCITypes /SLM\_ApplicableCITypes:** CIT that could use this BR.

**Example:**

*sid\_business\_interaction,monitor, location,sid\_product, sid\_resource, sid\_management\_domain, sid\_service,party, sid\_business\_interaction\_item.*

- (3) Do not set identifier **DASH\_Identifier / SLM\_Identifier**).



Do not set the **Identifier**. BSM will create one when you load the Content Pack.



Best Practice is to re-import those newly created BRs into SQM SD after


Importing CP in BSM

Exporting CP of BSM

Using the reverse Transducer.

- (4) Enter the location of the Java class that defines the Business Rule, in the **DASH\_JavaClass** (or **SLM\_JavaClass**) text field. This property will be mapped in BSM.


**Example:** com.hp.sqmbasm.businessrules.dashboard.generic.TestRule

- (5) Click  on the right side of value field, to select the type of the Business Rule, in the **DASH\_RuleType** (or **SLM\_RuleType**) text field. This property will be mapped in BSM.

DASH_RuleType	
DASH_Units	Health Indicator
DisplayName	KPI
SLM ApplicableCITypes	Sibling

8. Optional parameters:

- (1) **DASH\_Domain/SLM\_Domain:** SQM uses “telecom”.
- (2) **DASH\_RuleParameters/SLM\_RuleParameters:** To add **BR**

parameters click . On pop-up window click **<Add>**, and enter parameter with mandatory fields:

- Defaultvalue
- Description
- Mandatory
- Name
- Type
- isParamNameReadOnly
- Hidden

**Example:**

```
defaultvalue="" description="HI Type as defined in the repository"
mandatory="" name="HI2" type="String" isParamNameReadOnly="false"
hidden="false"
```



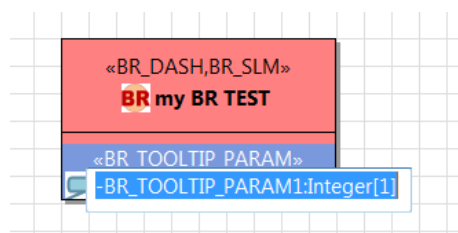
For more information about the mapping between CMDB properties and UML properties of different model components, refer to *Appendix A*.

For more information about different properties of the BSM components discussed above such as *Group Rule*, *Qualifiers*, *Label Format* etc., refer to the relevant *BSM documentation*.

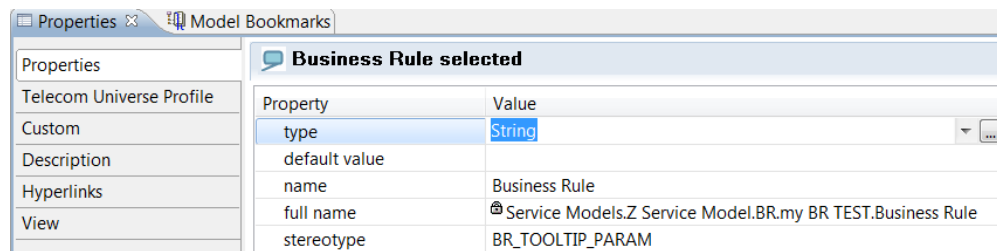
### 2.4.1.1 Defining a Business Rule Tool Tip Parameter

To define Business Rule Tooltip parameter in a UML model using the Telco Palette, you will be required to perform the following steps:

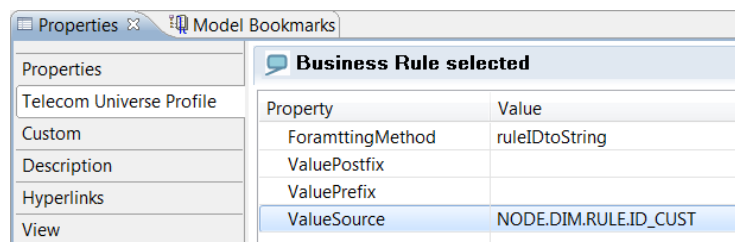
1. From the **Telco Service Designer Palette**, select the **BR\_TOOLTIP\_PARAM** stereotype.
2. Draw the object in the **Business Rule**.



3. Enter a parameter **name** and **type** in the **Properties** tab:



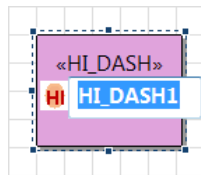
4. Enter the mandatory “**formattingMethod**” and “**ValueSource**” fields in the **Telecom Universe Profile** tab:



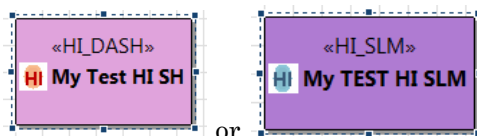
## 2.4.2 Defining an Health Indicator

To define HI in a UML model using the Telco Palette, you will be required to perform the following steps:

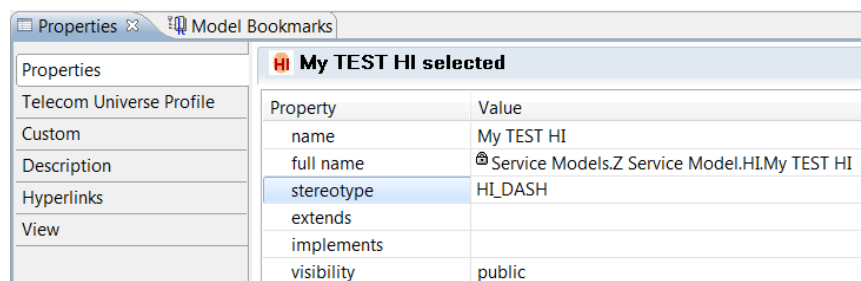
1. Select the **HI\_DASH** or **HI\_SLM** stereotype. Your choice should depend on whether you want to define a *Dashboard* or *SLM HI*.
2. Draw the object in the workspace.



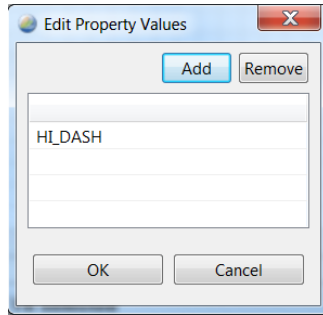
3. Enter a **name** (BSM “**Display Label**”):



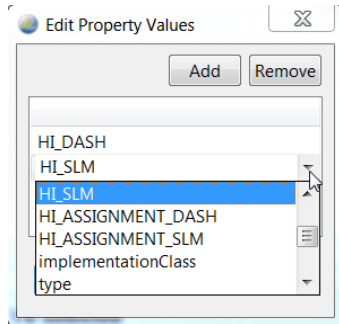
4. Select the newly created HI object. The **Properties** pane displays at the bottom of the main application window, and Verify the **name**:



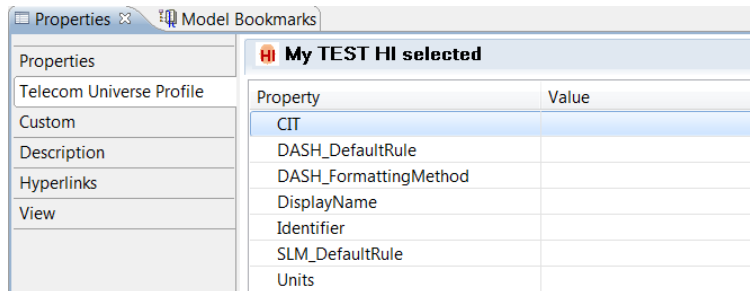
5. If the scope of HI is “**Service Health**” and “**SLM**”, then select “**stereotype**” value, and on the right side click
6. On the pop-up window, click <Add>



7. Use the drop down menu, to select “**HI\_SLM**”, and click <OK>



8. From the **Properties** pane, select **Telecom Universe Profile** tab, as shown in the figure below.



**Note:** Identifier Value will be generated by transducer.

**Note:** In case of “**Service Health**” HI, **SLM\_DefaultRule** is not present.

9. **Mandatory:**

- (1) **CIT:** BSM Name of CIT on which HI is attached.
- (2) **DisplayName:** BSM Name of HI

10. **Mandatory (DASH):**

- (1) **DASH\_DefaultRule:** the BSM **Service Health** Rule that does apply on this HI (Rule Type = “**Health Indicator**”).



use the field “Property > Name” of your BR.

Value is following this naming convention:

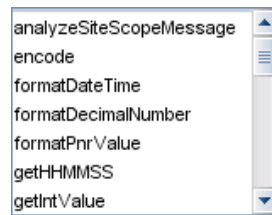
BR-<Name of the Business Rule>-(Service Health|SLM)

**Example:** BR-API Simplified Average Rule(SLM)

**Note:** please refer to BSM doc for more information : [Applications](#) > [Using Service Health](#) > [Repositories](#) > [Business Rule Repository](#) > [List of Calculation Rules in Service Health](#)

- (2) **DASH\_FormattingMethod**: enter the BSM Formatting Method.

For Example:



**Note:** please refer to BSM doc for more information : [Applications](#) > [Using Service Health](#) > [Repositories](#) > [KPI Repository](#) > [New KPI/Edit KPI Dialog Box](#) > [List of Formatting Methods](#)

#### 11. **Mandatory (SLM):**

- (1) **SLM\_DefaultRule**: enter the BSM **SLM** Rule that does apply on HI (Rule Type = “**Health Indicator** “):

**Note:** please refer to BSM doc for more information : [Applications](#) > [Using Service Level Management](#) > [Service Level Management Repositories](#) > [Business Rule Repository](#) > [List of Service Level Management Business Rules](#)

#### 12. **Optional:** Units: string to represent the HI units

#### 13. Do not set identifier **DASH\_Identifier** / **SLM\_Identifier**).



Do not set the **Identifier**. BSM will create one when you load the Content Pack.



Best Practice is to re-import those newly created HIs into SQM SD after

Importing CP in BSM

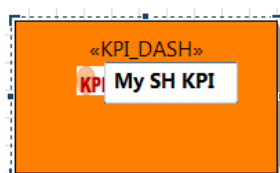
Exporting CP of BSM

Using the reverse Transducer.

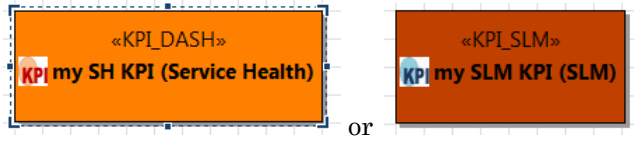
### 2.4.3 Defining a Key Performance Indicator

To define KPI in a UML model using the Telco Palette, you will be required to perform the following steps:

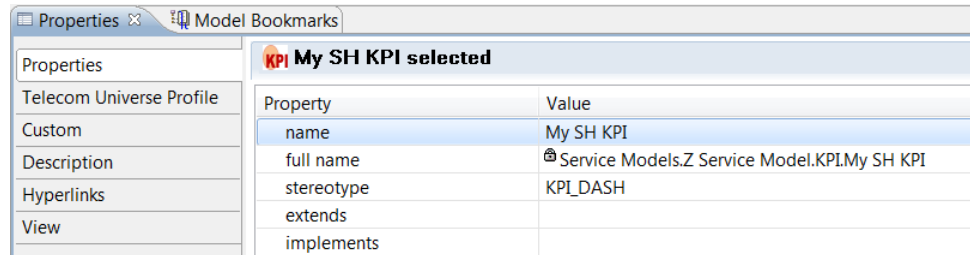
1. From the **Telco Service Designer Palette**, select the **KPI\_DASH** or **KPI\_SLM** stereotype. Your choice should depend on whether you want to define a *Dashboard KPI* or *SLM KPI*.
2. Draw the object in the workspace, and enter your new KPI name



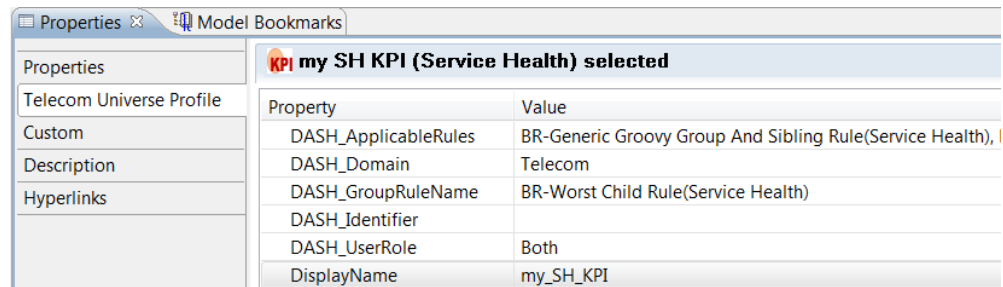
3. Select the newly created KPI object. The **Properties** pane displays at the bottom of the main application window.



4. Verify the KPI name in the **name** text field. The name will be mapped in BSM.



5. From the **Properties** pane, select **Telecom Universe Profile** tab, as shown in the figure below.



6. Enter a list of Applicable Rules: **ApplicableRules**,



use the field "Property > Name" of your BR.  
Name is following this naming convention:  
BR-<Name of the Business Rule>-(Service Health|SLM)  
**Example:** BR-API Simplified Average Rule(SLM)

**Note:** You could view the list of Calculation Rules in [Service Health in Applications > Using Service Health > Repositories > Business Rule Repository > List of Calculation Rules in Service Health](#)

7. Enter a **Domain**: for Example Telecom
8. Enter a default Group Rule Name from the list defined above: **GroupRuleName**,
9. Select User Role: **UserRole** (Both, Business, Operation)
10. Do not set identifier **DASH\_Identifier** / **SLM\_Identifier**).



Do not set the **Identifier**. BSM will create one when you load the Content Pack.



Best Practice is to re-import those newly created KPIs into SQM SD after  
Importing CP in BSM

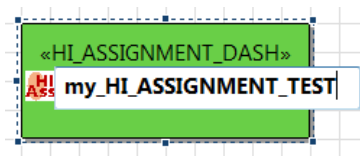


## 2.4.4 Defining an HI Assignment

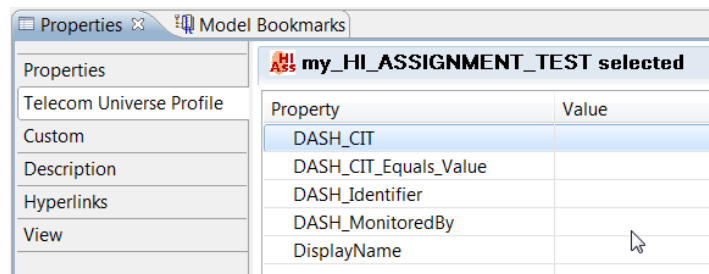
To define HI Assignment in a UML model using the Telco Palette, you will be required to perform the following steps:

1. From the **Telco Service Designer Palette**, select the **HI\_ASSIGNMENT\_DASH** or **HI\_ASSIGNMENT\_SLM** stereotype. Your choice should depend on whether you want to define a *Dashboard* or *SLM HI Assignment*.

2. Draw the object in the workspace, and enter your new HI Assignment name



3. Select the newly created HI\_Assignment object. The **Properties** pane displays at the bottom of the main application window.
4. Verify the HI\_Assignment name in the **name** text field. The name will be mapped in BSM.
5. From the **Properties** pane, select **Telecom Universe Profile** tab, as shown in the figure below.



6. Enter a CIT, **DASH\_CIT** or **SLM\_CIT**, to which the HI Assignment apply
7. **Optional:** Enter CIT Equals Value, **DASH\_CIT\_Equals\_Value** or **SLM\_CIT\_Equals\_Value**

In HI assignment definition, there will be two kinds of conditions

- `<condition cmdb-class="temip_managed_object" includeDerivedClasses="true" monitored-by="SQM">  
 <property-condition name="root_class" operator="EQ"  
 value="temip_managed_object"/>  
</condition>`
- `<condition cmdb-class="temip_service_console" includeDerivedClasses="true" monitored-by="SQM"/>`

8. Set the **MonitoredBy** to *SQM*
9. Enter a Display Name: **DisplayName**
10. For every HI, define an HI Configuration
11. Do not set identifier **DASH\_Identifier** / **SLM\_Identifier**).



Do not set the **Identifier**. BSM will create one when you load the Content Pack.



Best Practice is to re-import those newly created HI Assignments into SQM SD after

Importing CP in BSM

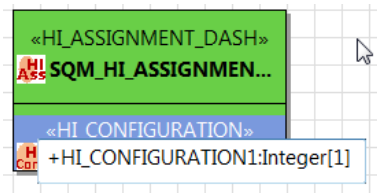
Exporting CP of BSM

Using the reverse Transducer.

### 2.4.4.1 Defining an HI Configuration

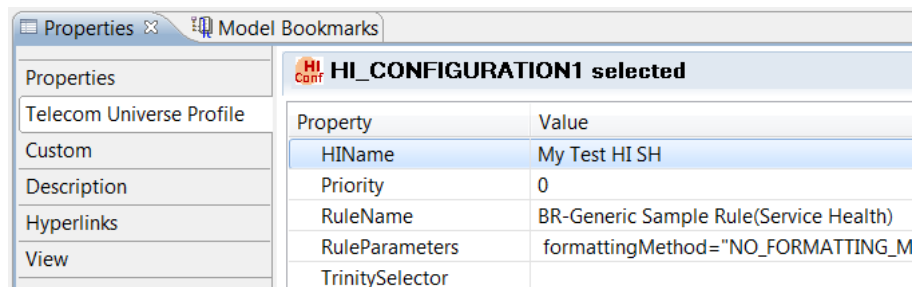
For every HI, you need to assign to your CI Type, you need to define an HI Configuration within HI assignment.

1. From the **Telco Service Designer Palette**, select the **HI\_CONFIGURATION** stereotype.
2. Draw the object in the workspace, and enter your new HI Configuration name



**Note:** you could keep the default or enter a meaning full name with your HI. Name just has to be unique within the HI Assignment.

3. Select the newly created HI\_Configuration object. The **Properties** pane displays at the bottom of the main application window.
4. Verify the HI\_Configuration name in the **name** text field. The name will be mapped in BSM.
5. From the **Properties** pane, select **Telecom Universe Profile** tab, as shown in the figure below.



6. Set the HI Name, **HIName** (from field of your HI: Property > Name)
7. Set a **priority** (default is 0)
8. Set the Rule Name, **RuleName**, used to retrieve HI value.
9. Set the Rule Parameters, **RuleParameters**. They depend of Rule Name.

For example on “Generic Sample Rule”, set

```
formattingMethod="NO_FORMATTING_METHOD" key="Time Stamp Field" type="String" value="time_stamp"
formattingMethod="NO_FORMATTING_METHOD" key="Field Name" type="String" value="dValue"
formattingMethod="NO_FORMATTING_METHOD" key="No data timeout" type="Long" value="900"
```

**Note:** You could view the list of Calculation Rules and their parameters in [Service Health in Applications](#) > [Using Service Health](#) > [Repositories](#) > [Business Rule Repository](#) > [List of Calculation Rules in Service Health](#)

10. Set the Selector, **TrinitySelector**, to find right CI.

For example if BSM exported CP file using eti\_id and szTagertName, it could be:

```
<CompositeSelector logicalOp="OR">
<CompositeSelector logicalOp="AND">
<Selector key="eti_id" op="EQ" type="Binary" value="60444f48d145487d8f172e927d61b4f8"/>
<Selector key="szTargetName" op="EQ" type="String" value="{RefProp:sid_object_identifier}"/>
<Selector key="sampleType" op="EQ" type="String" value="ss_t"/>
</CompositeSelector>
```

</CompositeSelector>Then we set values in Together Trinity Selector with three rows like this:

```
key="eti_id" op="EQ" type="Binary" value="60444f48d145487d8f172e927d61b4f8"
key="szTargetName" op="EQ" type="String" value="{RefProp:sid_object_identifier}"
key="sampleType" op="EQ" type="String" value="ss_t"
```



Trinity Selector could not be able to run after imprted into BSM, you might need to do a manually setup after the import to makr sure it works.

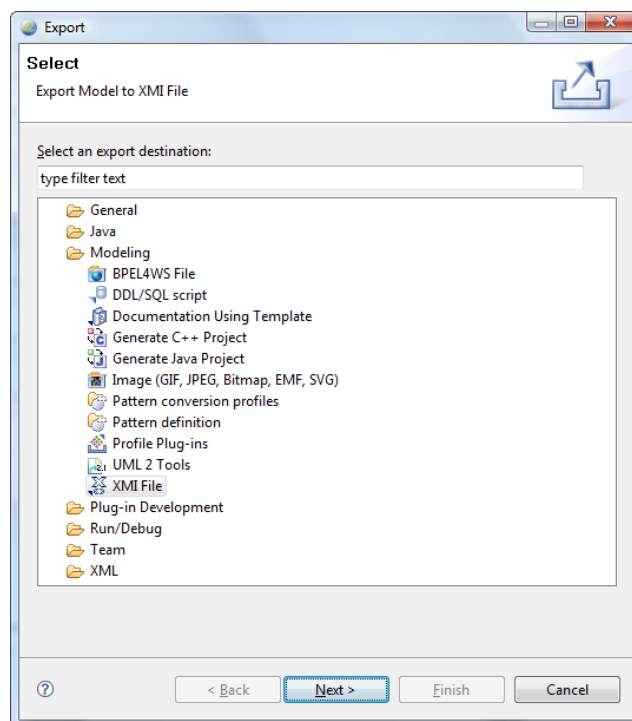
# Deploying Service Models in BSM

This chapter discusses the procedures involved in exporting UML models and converting these models to a BSM/RTSM-recognized service model and then importing them in *BSM*.

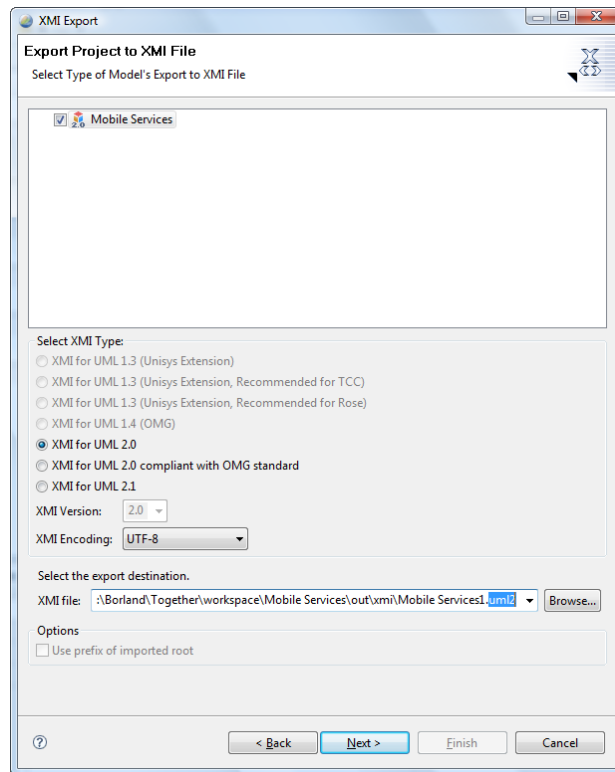
## 3.1 Exporting a UML Model

To export a UML model from *Together*, you will be required to perform the following steps:

1. Launch the *Together* application from the directory where it is installed
2. Open the project containing the UML model that you want to export
3. Select **File** → **Export...** to export the UML model
4. The Export window displays

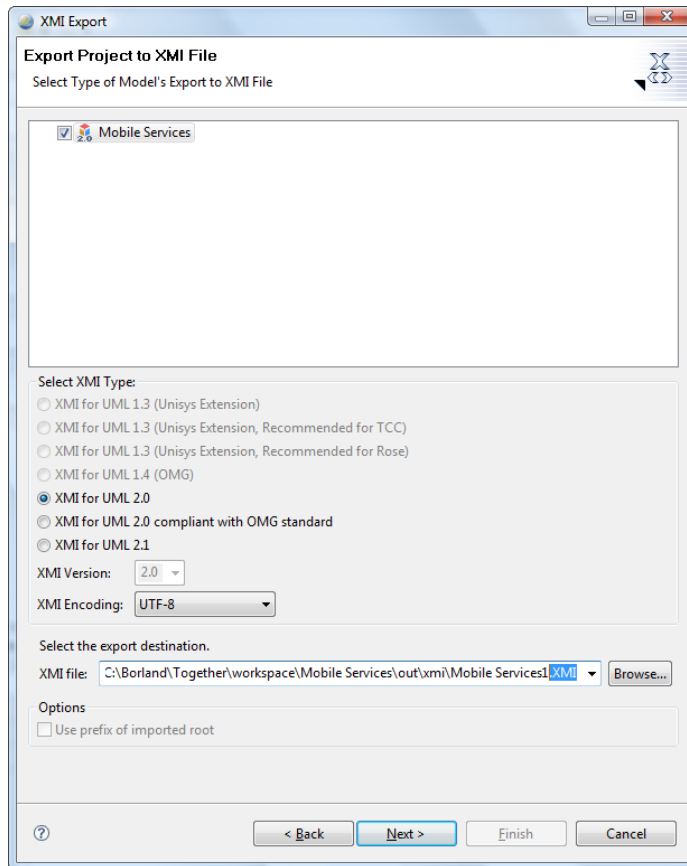


5. Select **XMI File** from the **Select an export destination** list and then click [Next]. The XMI Export window displays



6. Select **XMI for UML 2.0** from the **Select XMI Type** list

7. In the **Select the export destination: XMI file** text field, rename the extension of the output file from *UML2* to *XMI*, as shown in the figure below

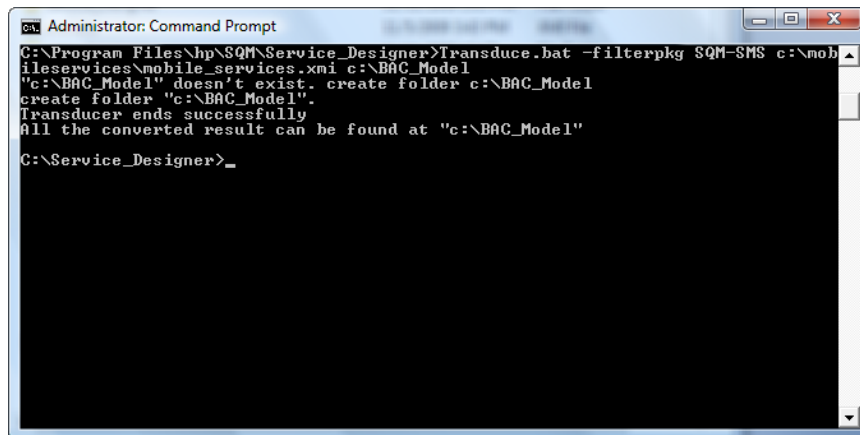


8. You can also change the destination path of the output .XMI file. Click [Finish] to complete the export procedure.
9. The UML model will be exported as an .XMI file to the selected destination folder.

## 3.2 Translating a UML Model

Once you have exported the UML model as an .XMI file, it can now be converted to a BSM/RTSM-recognized package and/or Content Pack. It is recommended to use the *UML Package* option in order to focus on a specific domain model. To translate a UML model .XMI file, you will be required to perform the following steps:

1. Launch the Windows Command Prompt from the **Start** menu and enter the Service Designer installation directory.
2. Execute the **Transduce.bat** command by entering required parameters i.e. *input file path* and *output folder path*. The input path must contain the .XMI file exported from *Together*. Use the mandatory *-filterpkg* option to convert a specific package in the .XMI file. You should use different UML packages for CIT/Relationships and KPIs/BRs. The name of the output .ZIP file will be same as the name of the UML package. An example of *Transducer* tool execution is shown in the figure below.



```
C:\Program Files\hp\SQM\Service_Designer>Transduce.bat -filterpkg SQM-SMS c:\mobileservices\mobile_services.xml c:\BAC_Model
"c:\BAC_Model" doesn't exist. create folder c:\BAC_Model
create folder "c:\BAC_Model".
Transducer ends successfully
All the converted result can be found at "c:\BAC_Model"
C:\Service_Designer>
```

Transduce.bat Usage:

C:\>Transduce [-options] <Together XMI file> <output folder>

where options include:

**-help** {print the help text}

**-filterpkg <package name>** {specify package in XMI file to be converted}

3. A message will be displayed confirming the successful conversion of the UML model file to a BSM/RTSM-recognized package and/or Content Pack. Depending on the contents of the selected UML package, a .ZIP file will be generated that contain definitions for CITs and relationships and a .XML file will be generated that contain definitions for BRs, HIs, and KPIs.



It is not recommended to modify the Telco Universe otherwise SMF will not be supported.

## 3.3 Deploying a Service Model in BSM

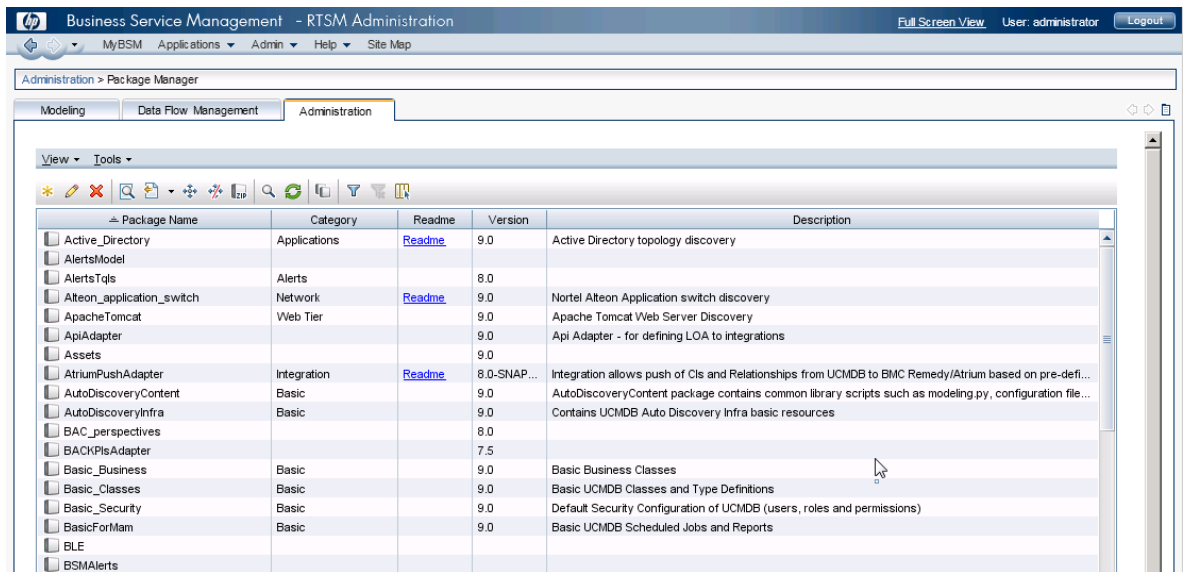
Once the UML model is successfully converted to a BSM/RTSM-recognized model, you can import it in *BSM* as a service model package. To perform these operations, you will be required to follow different procedures for deploying CITs and relationships (see ...).



If the UML package contains HI, KPI, BR, a Content Pack XML file will be generated and you will have to perform other operation to deploy HIs, KPIs, propagation rules and business rules.

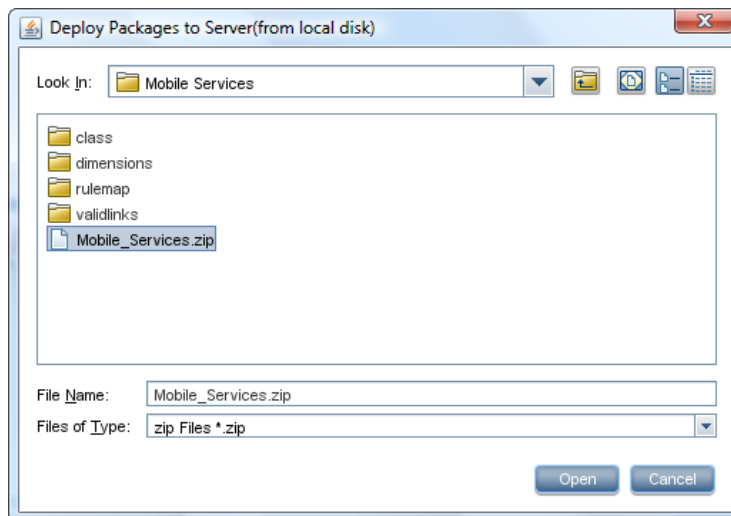
### 3.3.1 Deploying CITs and Relationships in BSM

To deploy CITs and relationships contained in a service model, in BSM, you will be required to perform the following steps:

1. Launch the *Business Availability Centre* (BSM) application. For information about launching and logging in to BSM, refer to the BSM documentation
2. Select **Admin → Universal CMDDB** and then click the **Settings → Package Manager** tab
3. The Settings > Package Manager window displays

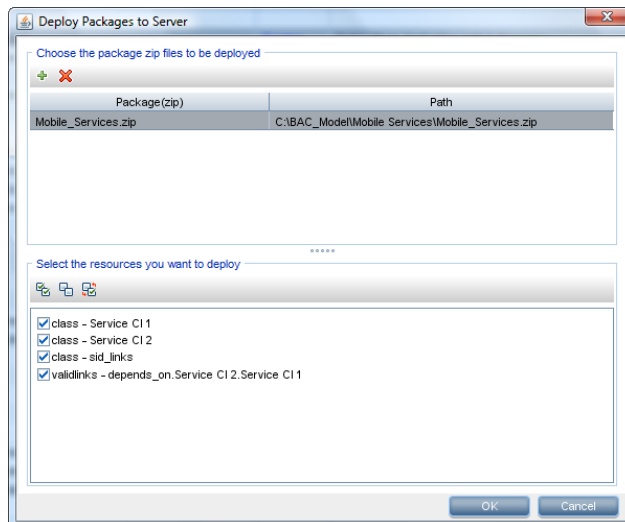


4. Click  [Deploy Packages to server (from local disk)]
5. The Deploy Packages to Server dialog displays
6. Click  [Add], browse and select the .ZIP package generated by the *Transduce.bat* command-line tool. An example is shown in the following figure



7. Once you add the .ZIP package, the Deploy Packages to Server dialog displays





8. From the **Select the resources you want to deploy** list, select the checkbox adjacent to the resources that you want to import into BSM.

**Class:** represent CI Types

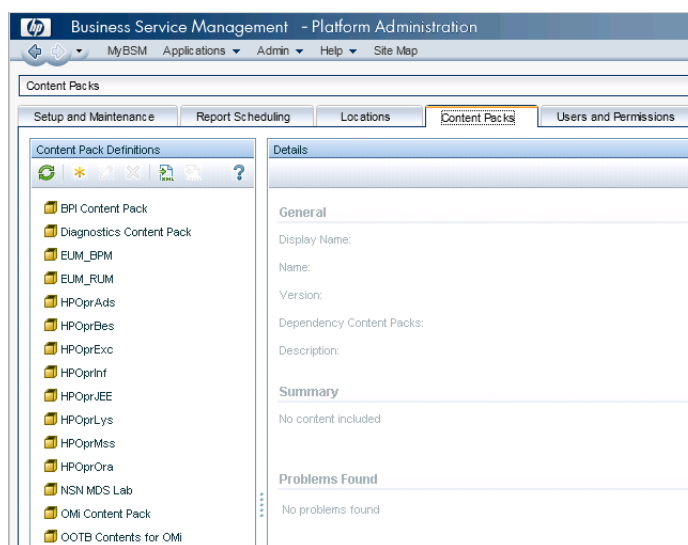
**validlinks:** represent relationships.


9. Click [OK] to import the selected package resources into BSM/RTSM.

### 3.3.2 Deploying BRs, HIs and KPIs in BSM

To deploy HIs, HI assignment, KPIs, KPI assignments, Propagation Rules and Business Rules contained in a service model, in BSM, you will be required to perform the following steps:

1. Launch the *Business Service Management* (BSM) application. For information about launching and logging in to BSM, refer to the BSM documentation
2. Select **Admin → Platform** and then click the **Content Packs** tab
3. The Content Packs Manager window displays



4. Click  icon to import new Content Pack

5. On the “Import Content Pack” window, select the CP XML file, check “**Create**” and “**Test**”

New\_BR\_and\_KPI.xml - Import Content Pack

Content Pack File: \* New\_BR\_and\_KPI.xml

☐ Overwrite  
Imports all objects contained within the content pack, including the content pack definition. Any objects existing in the target system with IDs that match objects in the specified content pack are overwritten. Any new objects are created. If any IDs do not match, the entire import is cancelled.

☒ Create  
Imports all objects contained within the content pack, including the content pack definition, and ignores any objects in the content pack, excluding the content pack definition, that currently exist on the target system.

☒ Test  
Run import in test mode only. No changes are committed to the database.

(\*) Required field

Import Cancel Help

6. Click “**Import**”, and If there is no error

Import New\_BR\_and\_KPI.xml - Result

**i** Import of Content Pack [New BR and KPI] was successfully tested. Content not modified.

Number of Business Rule objects successfully processed:	4/4.
Number of Content Pack Definition objects successfully processed:	1/1.

OK

7. Click “**OK**”, uncheck “**Test**”, and click “**Import**”.

## Chapter 4


# Importing Service Models in UML

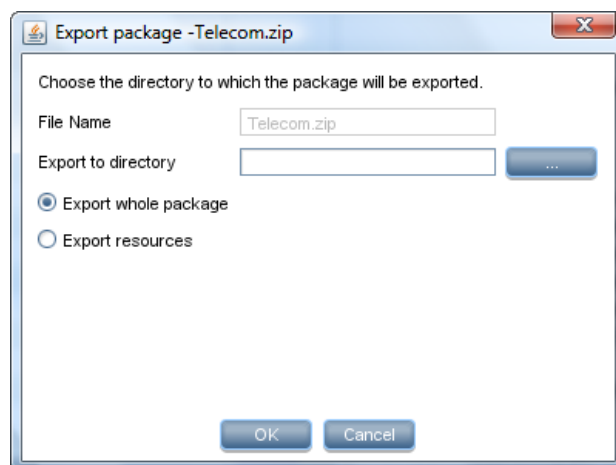
This chapter discusses the procedures involved in exporting BSM/RTSM service models and Content Packs, then converting them to a UML model and finally importing them in *Together* in a UML package.

## 4.1 Retrieving a Service Model

### 4.1.1 Exporting CITs and Relationships from BSM

To retrieve or export CITs and relationships in a service model from BSM/CMDB, you will be required to perform the following steps:

1. Launch the *Business Service Management* (BSM) application. For information about launching and logging in to BSM, refer to the BSM documentation
2. Select **Admin → RTSM Administration** and then click the **Administration → Package Manager** tab
3. The Package Manager window displays
4. Select your RTSM package
5. Click  [Export Packages to local disk]



6. Select the destination directory by clicking [...] and then click [OK] to export the service model package to the selected destination.




You will need to export the *Telco Universe* model if any other model has a relationship with the Telco Universe model. To export the Telco Universe model, you will be required to select **Telecom** package from the Package Manager list.

The service model package is exported as a *.ZIP* format file.

### 4.1.2 Exporting Content Pack from BSM

To retrieve or export HIs, HI assignments, KPIs, Business Rules, and Business Rules in a Content Pack from BSM/RTSM, you will be required to perform the following steps:

1. Launch the *Business Service Management* (BSM) application. For information about launching and logging in to BSM, refer to the BSM documentation
2. Select **Admin** → **Platform** and then click the **Content Packs** tab
3. The Content Packs Manager window displays
4. Select your Content Pack
5. Edit it to verify its contents
6. Click  [Export Content Pack Definition and Content]
7. Specify the directory and File name (.XML)
8. Click [SAVE]

## 4.2 Translating a Service Model

A Reverse Transducer can convert BSM service model or content pack files into UML recognized model which is used in Together. Following previous guide, you have retrieved all the required components of the service model as *.ZIP* files, and Content Packs as *.XML* file, then move them to the one folder which will be *input folder* of the *ReverseTransducer.bat* command-line tool. An example of the *Reverse Transducer input folder* is shown in the following figure:



ReverseTransduce.bat Usage:

**ReverseTransduce**

-folder/-f <BSM\_XML\_directory> -target/-t <Together XMI file>

OR

**ReverseTransduce**

-zipfile/-z <zipfile1.zip; ...;zipfileN.zip> -cp  
<content\_pack\_file1.xml;...;content\_pack\_fileN.xml> -target/-t  
<Together XMI file>

OR

**ReverseTransduce**

-zipfile/-z <zipfile1.zip;...;zipfileN.zip> -target/-t <Together XMI  
file>

OR

**ReverseTransduce**

-cp <content\_pack\_file1.xml;...;content\_pack\_fileN.xml> -target/-t  
<Together XMI file>

The parameter explanation:

**[-folder/-f <BSM\_XML\_directory>]**: a folder as the input of the command, it mean all resource package zip files and content package are put under this folder, the structure should be:

```
--folder
  --model1.zip
  --model2.zip
  --...
  --modelN.zip
  --cp1.xml
  --cp2.xml
  --...
  --cpN.xml
```

**-zipfile/-z <zipfile1.zip;...;zipfileN.zip>**: specify the zip files explicitly, multi files should be separated by semicolon

**-cp <content\_pack\_file1.xml;...;content\_pack\_fileN.xml>**: specify the content files explicitly, multi files should be separated by semicolon.

You can just specify the zip files or cp files or both of them.

**-target/-t <Together XMI file>**: mandatory, the output file.

Here are some examples:

```
>ReverseTransduce -folder /test -target mytest.xmi
>ReverseTransduce -zipfile /test/sid.zip;/test/mgt.zip -cp
qadvisor.xml;temip.xml -target mytest.xmi
>ReverseTransduce -z /test/sid.zip;/test/mgt.zip -target mytest.xmi
>ReverseTransduce -cp qadvisor.xml;temip.xml -t mytest.xmi
>ReverseTransduce -z /test/sid.zip;/test/mgt.zip -cp
qadvisor.xml;temip.xml -t mytest.xmi
```

The output of this command is an xmi file, and after you import the xmi file into a together UML2.0 project, the structure should be as below:

Under the project, two top level package will be created: Service Models, Service Repository, and under the service model package, build one sub package for each zip file, and under the Service Repository package, build one sub package for each content pack file. Here is an example:

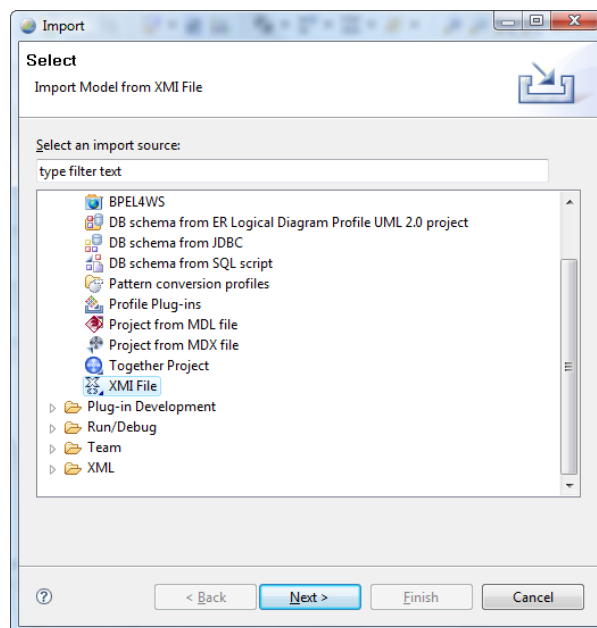
```
--Project
  --Service Models
    --Shared Information Model
      --<CIT Hierarchy>
    --TeMIP Management Modle
      --<CIT Hierarchy>
  --Service Repository
    --QAdvisor
      -- <HI, KPIs ...>
    --TeMIP
      -- <HI, KPIs ...>
```

**Note:** currently, all CIT in one zip will be put under the relevant package in the same level.

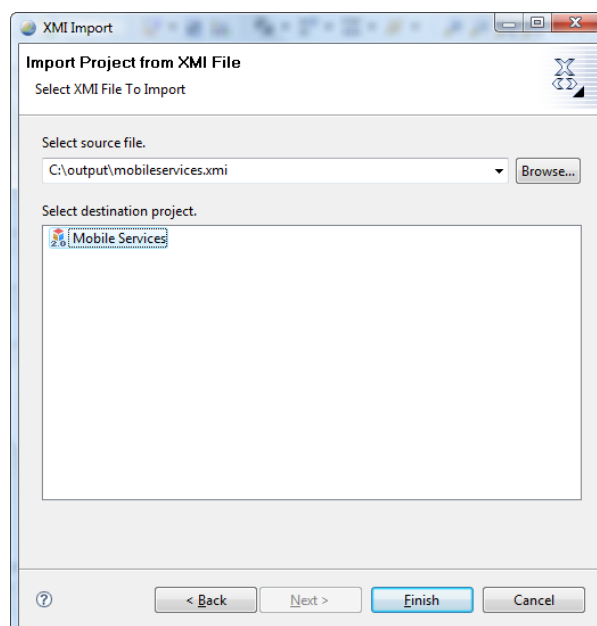
## 4.3 Importing and Viewing a Service Model in UML

Once the BSM/RTSM service model is successfully converted to a UML-recognized model, you can import it in *Together* in a UML package and view it as a UML model. To import and view a converted service model in *Together*, you will be required to perform the following steps:

1. Launch *Together* to import and view the converted service model.
2. Create a new UML project. For information about creating a new UML project, refer to *1.1.1 Enabling the Telecom Universe Profile*.
3. Select **File** → **Import**. The Import window displays.



4. Select **XMI File** as the import source and then click [Next]
5. The XMI Import window displays.



6. Click [Browse] and browse to the location of the .XMI file generated by the *Reverse-Transducer* procedure.
7. Select an existing project from the **Select destination project** list. The imported model will be saved in the selected project.
8. Click [Finish] to complete the import and view the service model.

# Troubleshooting and License Management

This chapter deals with different aspects of troubleshooting during the course of performing various operations using the Service Designer solution. It consists of detail on how to configure Service Designer to complement troubleshooting procedures. This chapter also explains the procedure of installing or updating official license for Service Designer.

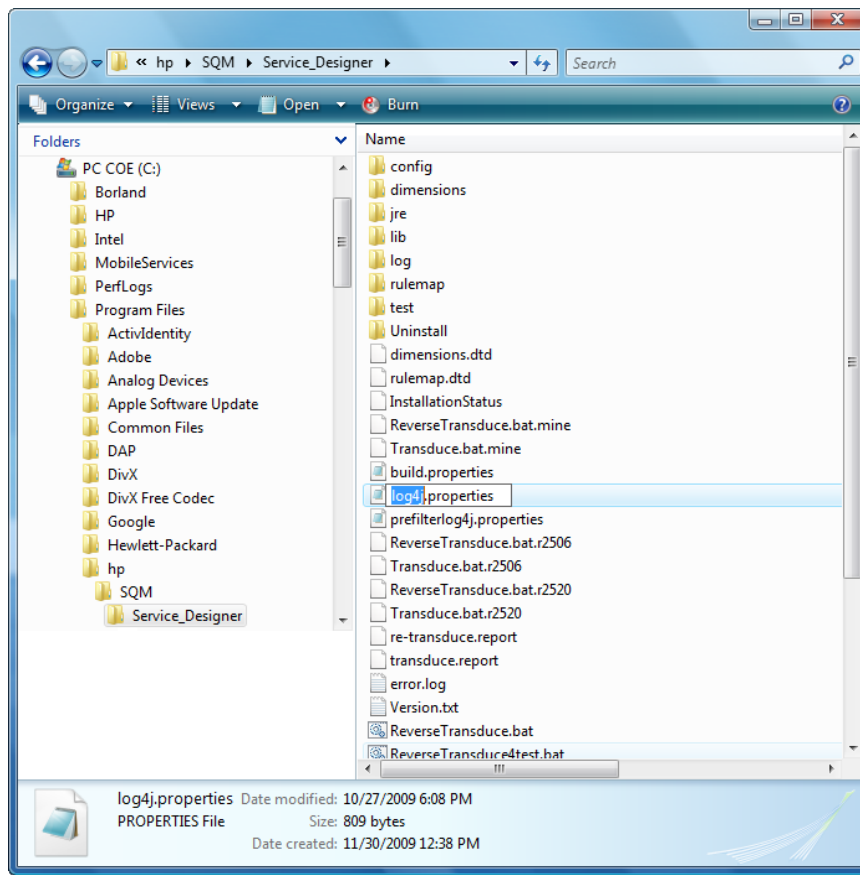
## 5.1 Configuring Error Tracing

To configure the error tracing mechanism which is activated during the use of *Transducer* and *Reverse-Transducer* procedures to log various types of errors, you will be required to perform the following steps:

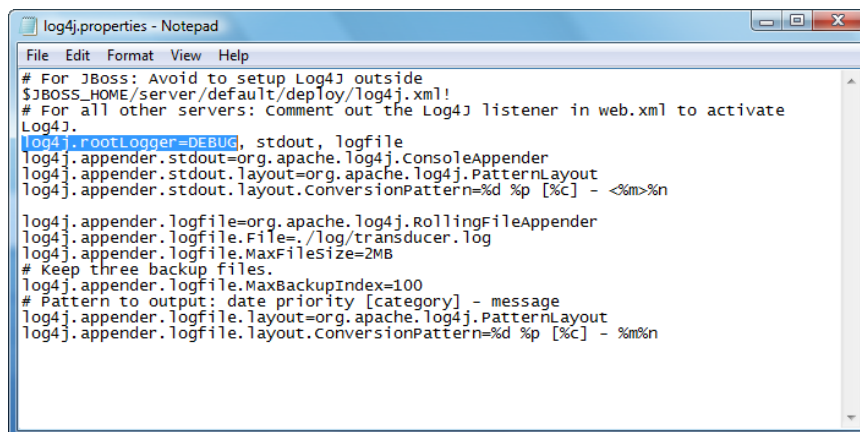
1. Browse to the location on your system where you have installed *Service Designer*.



2. The following figure shows an example of the default installation path.



3. Open **Log4j.properties** file in a suitable editor (such as, MS Notepad).
4. The content of the file displays in the editor window. An example is shown in the figure below.



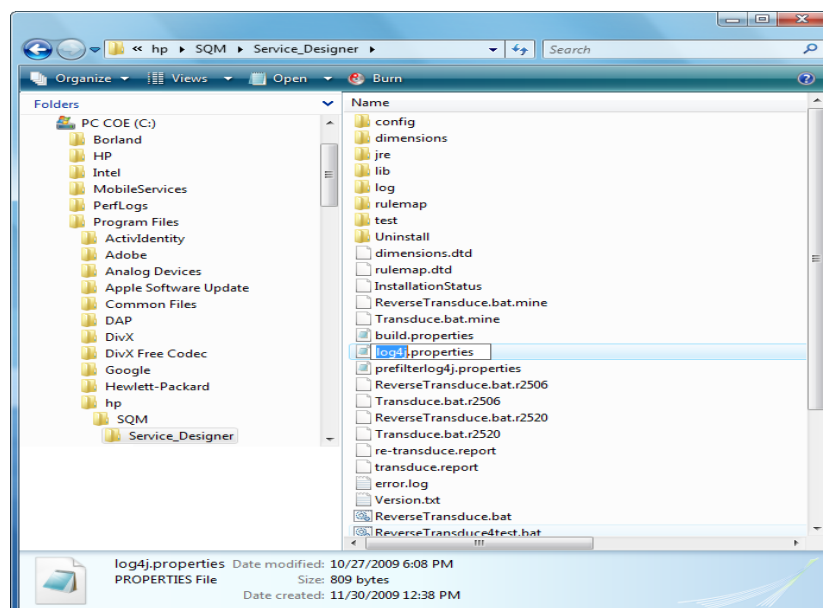
5. You can configure the level of traces for the *Transducer* and *Reverse-Transducer* procedures by editing the **log4j.rootLogger** property. By default the level is set to **ERROR**. You can change this value to one of the following levels, as appropriate:

Level	Description
<b>FATAL</b>	Severe errors that cause premature termination. Expect these to be immediately visible on a status console.
<b>ERROR</b>	Other runtime errors or unexpected conditions. Expect these to be immediately visible on a status console.
<b>WARN</b>	Use of deprecated APIs, poor use of API, 'almost' errors, other runtime situations that are undesirable or unexpected, but not necessarily "wrong". Expect these to be immediately visible on a status console.
<b>INFO</b>	Interesting runtime events (startup/shutdown). Expect these to be immediately visible on a console, so be conservative and keep to a minimum.
<b>DEBUG</b>	Detailed information on the flow through the system. Expect these to be written to logs only.
<b>TRACE</b>	More detailed information. Expect these to be written to logs only.

## 5.2 Configuring Log Generation

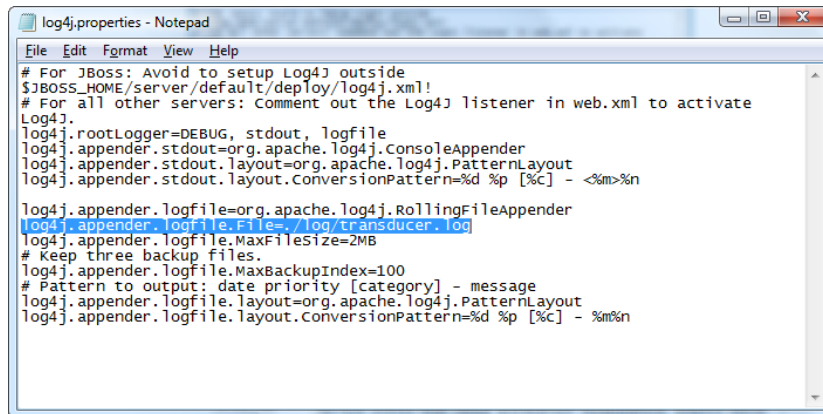
To configure the location where the error log will be generated during execution of the *Transducer* and *Reverse Transducer* procedures, you will be required to perform the following steps:

1. Browse to the location on your system where you have installed *Service Designer*. The following figure shows an example of the default installation path.



2. Open **Log4j.properties** file in a suitable editor (such as, MS Notepad).

3. The content of the file displays in the editor window. An example is shown in the figure below.



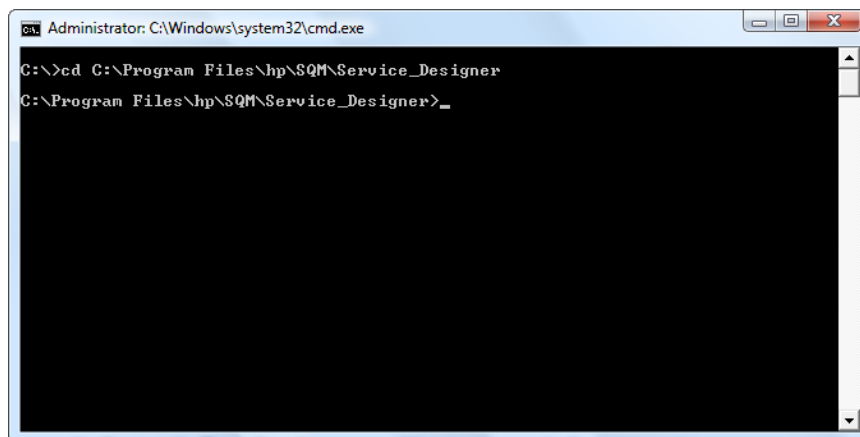
```
# For JBoss: Avoid to setup Log4j outside
$JBOSS_HOME/server/default/deploy/log4j.xml!
# For all other servers: Comment out the Log4j listener in web.xml to activate
Log4j
log4j.rootLogger=DEBUG, stdout, logfile
log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern=%d %p [%C] - <%m>%n
log4j.appender.logfile=org.apache.log4j.RollingFileAppender
log4j.appender.logfile.File=./log/transducer.log
log4j.appender.logfile.MaxFileSize=2MB
# Keep three backup files.
log4j.appender.logfile.MaxBackupIndex=100
# Pattern to output: date priority [category] - message
log4j.appender.logfile.layout=org.apache.log4j.PatternLayout
log4j.appender.logfile.layout.ConversionPattern=%d %p [%C] - %m%n
```

4. You can edit the value of the **log4j.appender.logfile.File** property, as appropriate. For example, if this property is set to **./log/transducer.log**, it means that the error log will be saved as **transducer.log** in a new folder called **log** which will be created in the current installation folder of *Service Designer*.

## 5.3 Managing Licenses

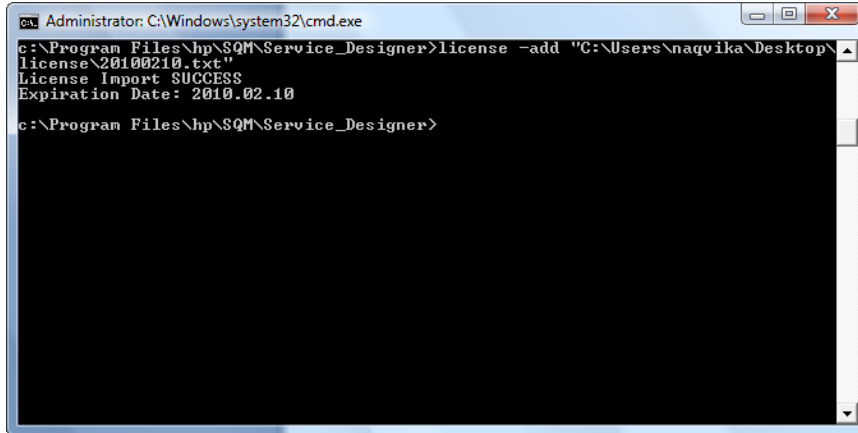
To install or update a new license for Service Designer, you will be required to perform the following steps:

1. Launch the Windows Command Prompt from the **Start** menu and enter the Service Designer installation directory. An example is shown in the figure below.



```
Administrator: C:\Windows\system32\cmd.exe
C:\>cd C:\Program Files\hp\SQM\Service_Designer
C:\Program Files\hp\SQM\Service_Designer>_
```

2. Execute the **License.bat** command by entering required parameters i.e. *-add* *<license file path>*. The license file path must contain a valid license file. An example is shown in the figure below.



```
Administrator: C:\Windows\system32\cmd.exe
c:\Program Files\hp\SQM\Service_Designer>license -add "C:\Users\naquika\Desktop\
license\20100210.txt"
License Import SUCCESS
Expiration Date: 2010.02.10
c:\Program Files\hp\SQM\Service_Designer>
```

3. Press [Enter] to execute the *License.bat* tool, which will install the selected license for Service Designer and show a confirmation message along with the date of expiration for the installed license.



License.bat Usage:

C:>License [-options]

where options include:

*-help* {print the help text}

*-add <license file path>* {specify the location of the license file to be imported}

*-GUI* {Pop a GUI to perform the license management operations}.

# Appendix A

This section contains the mapping information for different components of a service model. For each component, a separate table defines the mapping of applicable properties and their names in BSM (RTSM/BSM Repositories) and UML.

## CI Type

Property in CMDB	Property in Together UML	Type
Display Name	Properties -> Name	String
Derived-From class-name	Properties -> Extends	String
Name	Telecom Universe Profile -> Identifier	String
Qualifiers	Telecom Universe Profile -> Qualifiers	String
Icon	Telecom Universe Profile -> Icon Name	String
Default Label	Telecom Universe Profile -> Label Format	String
Description	Description	String

## CIT Attribute

Property in CMDB	Property in Together UML	Type
Attribute Name	Telecom Universe Profile -> Identifier	String
Display Name	Properties -> name	String
Type	Properties -> type	String
Default Value	Properties -> default value	String
Editable (=false)	Properties -> read only	Boolean
Static (=true)	Properties -> static	Boolean
Unique (=true)	Properties -> is unique	Boolean
Index (=true)	Telecom Universe Profile -> Index	Boolean
Key (=true)	Telecom Universe Profile -> Key	Boolean
Required (=true)	Telecom Universe Profile -> Required	Boolean
Visible (=false)	Telecom Universe Profile -> Visible	Boolean

Property in CMDB	Property in Together UML	Type
Asset Data (=true)	Telecom Universe Profile -> Asset Data	Boolean
Comparable	Telecom Universe Profile -> Comparable	Boolean
Change Monitored (=true)	Telecom Universe Profile -> Change Monitored	Boolean
Enable Auto Truncate by DDM (=true)	Telecom Universe Profile -> Enable Auto Truncate by DDM	Boolean
Upper Case (=true)	Telecom Universe Profile -> Upper Case	Boolean
Lower Case (=true)	Telecom Universe Profile -> Lower Case	Boolean
Password (=true)	Telecom Universe Profile -> Password	Boolean
Value Size	Telecom Universe Profile -> Size	Integer
Description	Description	String

## CIT Attribute Override

Property in CMDB	Property in Together UML	Type
Attribute Name	Telecom Universe Profile -> Identifier	String
Default Value	Properties -> default value	String
Editable (=false)	Properties -> read only	Boolean
Static (=true)	Properties -> static	Boolean
Unique (=true)	Properties -> is unique	Boolean
Index (=true)	Telecom Universe Profile -> Index	Boolean
Key (=true)	Telecom Universe Profile -> Key	Boolean
Required (=true)	Telecom Universe Profile -> Required	Boolean
Visible (=false)	Telecom Universe Profile -> Visible	Boolean
Asset Data (=true)	Telecom Universe Profile -> Asset Data	Boolean
Comparable	Telecom Universe Profile -> Comparable	Boolean
Change Monitored (=true)	Telecom Universe Profile -> Change Monitored	Boolean
Enable Auto Truncate by DDM (=true)	Telecom Universe Profile -> Enable Auto Truncate by DDM	Boolean
Upper Case (=true)	Telecom Universe Profile -> Upper Case	Boolean
Lower Case (=true)	Telecom Universe Profile -> Lower Case	Boolean
Password (=true)	Telecom Universe Profile -> Password	Boolean

Property in CMDB	Property in Together UML	Type
Value Size	Telecom Universe Profile -> Size	Integer

## CMDB Relationship

Property in CMDB	Property in Together UML	Type
Class Ref class-name	Telecom Universe Profile -> Identifier	String
End1 class-name		String
End2 class-name		String
Valid-Link-Qualifiers	Telecom Universe Profile -> Qualifier	String

## CMDB Relationship Link Triplet

Property in CMDB	Property in Together UML	Type
class-name	Telecom Universe Profile -> Identifier	String
Calculated-link-type (='impact_dependency')	Telecom Universe Profile -> CalculatedLinkType	String
is-forward (=true)	Telecom Universe Profile -> IsForward	Boolean
End1 class-name		String
End2 class-name		String

## Business Rules

Property in CMDB	Property in Together UML	Type
ID	Telecom Universe Profile -> (DASH/SLM)_Identifier	Integer
Display Label	Properties -> Name	String
Description	Description	String
RuleType	Telecom Universe Profile -> (DASH/SLM)_RuleType	String
Class Name	Telecom Universe Profile -> (DASH/SLM)_JavaClass	String
Units	Telecom Universe Profile -> (DASH/SLM)_Units	String

## Health Indicators

Property in CMDB	Property in Together UML	Type
N/A	Properties-->name	String
application	Properties-->stereotype	String
description	Description	String
artifact id (generated by transducer)	Telecom Universe Profile -> identifier	String
ciTypeId	Telecom Universe Profile -> CIT	String
displayName, Label, name	Telecom Universe Profile --> DisplayName	String
units	Telecom Universe Profile --> Units	String
ServiceHealthIndicatorData defaultRule	Telecom Universe Profile --> DASH_DefaultRule	String
formattingMethod	Telecom Universe Profile --> DASH_FormattingMethod	String
SLMIndicatorData defaultRule	Telecom Universe Profile --> SLM_DefaultRule	String

## HI Assignment

Property in CMDB	Property in Together UML	Type
assignmentId	HI Assignment for Dashboard: Telecom Universe Profile --> DASH_ Identifier HI Assignment for SLM: Telecom Universe Profile --> SLM_ Identifier	String
displayName	Properties-->name	String
description	Properties-->Description	String
assignmentApplication Type	Properties-->stereotype	String
condition CIT	HI Assignment for Dashboard: Telecom Universe Profile --> DASH_ CIT HI Assignment for SLM: Telecom Universe Profile --> SLM_ CIT	String
condition Monitored-by	HI Assignment for Dashboard: Telecom Universe Profile --> DASH_ MonitoredBy HI Assignment for SLM: Telecom Universe Profile --> SLM_ MonitoredBy	String



## HI Assignment Configuration

Property in CMDB	Property in Together UML	Type
indicatorId	Telecom Universe Profile --> HIName	String
rule	Telecom Universe Profile --> RuleName	String
rule-parameter	Telecom Universe Profile --> RuleParameters	String
trinity-selector	Telecom Universe Profile --> TrinitySelector	String

## Key Performance Indicators

Property in CMDB	Property in Together UML	Type
ID	Telecom Universe Profile -> (DASH/SLM)_Identifier	Integer
Display Label	Properties -> Name	String
Applicable Sections	(depends on the KPI stereotype = DASH or SLM)	String
Default Group Rule	Telecom Universe Profile -> (DASH/SLM)_GroupRuleId	String
applicableRules	Telecom Universe Profile -> (DASH/SLM)_ApplicableRules	String
ID	Telecom Universe Profile -> (DASH/SLM)_Identifier	Integer

---

## A

associated  
documents ·6  
Associated documents ·6

---

## B

BR tooltip  
defining ·28  
BRs, HIs and KPIs  
Deploying ·41  
BSM to Together  
mapping ·53  
business rule  
defining ·26

---

## C

Calculated Link triplet  
Defining ·23  
CI type  
defining ·17  
CIT and relationships  
deploying ·39  
CIT attribute  
defining ·20  
CITs and relationships  
exporting ·43  
configuring  
error tracing ·48  
log generation ·50  
Content Pack  
Creating ·25  
Exporting ·44  
creating  
service model ·17  
Creating  
Content Pack ·25

---

## D

defining  
BR Tool Tip ·28  
business rule ·26  
CI type ·17  
CIT attribute ·20  
inheritance relationship ·22  
KPI ·31  
relationship ·21  
Defining

Calculated Link triplet ·23  
HI ·29  
HI Assignment ·33  
HI Configuration ·34  
deploying  
CIT and relationships ·39  
service models ·36  
designing  
service models ·12

---

## E

enabling  
telecom universe profile ·13  
error tracing  
configuring ·48  
exporting  
CITs and relationships ·43  
UML model ·36  
Exporting  
Content Pack ·44

---

## H

HI  
Defining ·29  
HI Assignment  
Defining ·33  
HI Configuration  
Defining ·34

---

## I

importing  
service models ·43  
Importing  
SMF UML workspace ·16  
importing and viewing  
service model ·46  
inheritance relationship  
defining ·22

---

## K

KPI  
defining ·31

---

## L

licenses

- managing ·51
- log generation
  - configuring ·50

---

## **M**

- managing
  - licenses ·51
- mapping
  - BSM to Together ·53

---

## **O**

- overview
  - service designer ·8

---

## **R**

- relationship
  - defining ·21
- retrieving
  - service model ·43
- reverse transducer ·9
  - workflow ·9

---

## **S**

- service designer
  - components ·8
  - overview ·8
  - workflows ·9
- service model
  - creating ·17
  - importing and viewing ·46
  - retrieving ·43

- translating ·44
- service models
  - deploying ·36
  - designing ·12
  - importing ·43
- SMF UML workspace
  - Importing ·16

---

## **T**

- telco palette ·8
  - using ·12
- telecom universe profile
  - enabling ·13
- transducer ·9
  - workflow ·11
- translating
  - service model ·44
  - UML model ·38
- troubleshooting ·48

---

## **U**

- UML model
  - exporting ·36
  - translating ·38
- using
  - telco palette ·12

---

## **W**

- workflow
  - reverse transducer ·9
  - transducer ·11