

Collaborative Systems Lifecycle Management & Traceability

- CS9 -



Implementation Guide – V1

BPA Delivery 7 for V5R19 (V5.7)

Modification Tracking

Version	Date	Done by	Modification
	06 Oct 2008	CBZ	Document creation
D6W42.3	15 Oct 2008	NJZ/CBZ	Document verification
D7W22.3	14 May 2009	CBZ	Adding Requirements IVVQ Traceability Matrix

Table of Contents

1. INTRODUCTION	7
1.1. Scope and purpose	7
1.2. Related Documentation	8
1.3. Definitions.....	8
1.3.1. Glossary	8
2. COLLABORATIVE SYSTEMS LIFECYCLE MANAGEMENT & TRACEABILITY IMPLEMENTATION OVERVIEW	9
2.1. How to hook script functions	9
2.2. Link Control only Script Hooking.....	11
2.2.1. System tab.....	11
2.2.2. User Defined tools tab	12
3. COLLABORATIVE SYSTEMS LIFECYCLE MANAGEMENT & TRACEABILITY HISTORY MANAGEMENT	13
3.1. System tab.....	13
3.1.1. General case.....	13
3.1.2. Specific case : Requirements class.....	14
3.2. User Defined tab	14
4. COLLABORATIVE SYSTEMS LIFECYCLE MANAGEMENT & TRACEABILITY CONFIGURING BOTH HISTORY MANAGEMENT AND LINK CONTROL.....	16
5. COLLABORATIVE SYSTEMS LIFECYCLE MANAGEMENT & TRACEABILITY LOOKUP TABLES	17
6. COLLABORATIVE SYSTEMS LIFECYCLE MANAGEMENT & TRACEABILITY ADDING ADMIN SETTINGS FOR LINKCONTROL.....	19
7. COLLABORATIVE SYSTEMS LIFECYCLE MANAGEMENT & TRACEABILITY UNDELETE	20
7.1. Adding attribute.....	20
7.2. Configure “Script Maintenance”.....	20

8. COLLABORATIVE SYSTEMS LIFECYCLE MANAGEMENT & TRACEABILITY VERIFICATION MATRIX.....23

8.1. Default verification matrixes	23
8.1.1. RVM matrix	23
8.1.1.1. Hook	23
8.1.1.2. Icon in toolbar	24
8.1.1.3. Admin settings	24
8.1.2. RAM matrix	27
8.1.2.1. Hook	27
8.1.2.2. Icon in toolbar	27
8.1.2.3. Admin settings	27
8.1.3. RCM matrix	29
8.1.3.1. Hook	29
8.1.3.2. Icon in toolbar	29
8.1.3.3. Admin settings	29
8.2. Adding user customized verification matrix.....	30

9. COLLABORATIVE SYSTEMS LIFECYCLE MANAGEMENT & TRACEABILITY TRACEABILITY MATRIX.....31

9.1. Default traceability matrixes	31
9.1.1. Requirements to Requirements traceability matrix	32
9.1.1.1. Hook	32
9.1.1.2. Icon in toolbar	32
9.1.1.3. Admin settings	32
9.1.2. Requirements to Functions traceability matrix (unique identifier = 2)	42
9.1.2.1. Hook	42
9.1.2.2. Icon in toolbar	42
9.1.2.3. Admin settings	42
9.1.3. Functions to Systems traceability matrix	43
9.1.3.1. Hook	43
9.1.3.2. Icon in toolbar	43
9.1.3.3. Admin settings	43
9.1.4. Requirements to IVVQ traceability matrix	44
9.1.4.1. Hook	44
9.1.4.2. Icon in toolbar	44
9.1.4.3. Admin settings	44
9.2. Adding user customized traceability matrix.....	45

10. APPENDIX A – ADMIN SETTINGS RULES46

10.1. Settings Folder.....	46
10.2. Avoid doubles	46
10.3. Remove the unused Admin Settings	47

List of figures and tables

Table 1 - Glossary	8
--------------------------	---

Copyright Notice

Copyright©2010. Dassault Systèmes, All Rights Reserved.

This guide is delivered subject to the following conditions and restrictions:

CONFIDENTIAL - This document contains unpublished, confidential and proprietary information of Dassault Systèmes.

This document or any part thereof shall not be reproduced or transferred to other documents or formats, disclosed to others or used for any purpose other than that for which it is furnished, without the prior written consent of Dassault Systèmes.

It shall be returned to Dassault Systèmes upon request.

Dassault Systèmes is a registered trademark of Dassault Systèmes.

All other trademarks belong to their respective owners.

ENOVIA SmarTeam is a registered trademark of Dassault Systèmes.

Microsoft Windows and Windows XP are registered trademarks of Microsoft Corporation in the United States and/or other countries.

1. Introduction

This document describes the implementation guide for the BPA CS9.

This document is divided into the following sections:

- Collaborative Systems Lifecycle Management & Traceability implementation overview
- Collaborative Systems Lifecycle Management & Traceability History Management
- Collaborative Systems Lifecycle Management & Traceability Configuring both History Management and Link Control
- Collaborative Systems Lifecycle Management & Traceability Lookup Tables
- Collaborative Systems Lifecycle Management & Traceability Adding Admin Settings for LinkControl
- Collaborative Systems Lifecycle Management & Traceability Undelete
- Collaborative Systems Lifecycle Management & Traceability Verification matrix
- Collaborative Systems Lifecycle Management & Traceability Traceability matrix
- Appendix A – Admin Settings rules

1.1. *Scope and purpose*

When manipulating links in SmarTeam, you have the possibility to activate traceability services. There are 2 of them:

- **History management**
- **Link Control**

History Management enables a traceability of modifications made on a database.

The Link Control can be used to justify a modification that would be done on an object in a database. This object can be for example a requirement or a link between 2 objects.

You can choose to use **either** History management **or** Link control **or both**. The configuration of these functions is done through the Script maintenance administrative tool in SmarTeam CSE.

- Functions related to History management are located in the Script named CSE_HA.bs.
- Functions related to Link Control are located in CSE_CMN.bs.
- Functions to set both services are either in CSE_HA.bs or CSE_CMN.bs.
- Functions related to History management and other services are in Common.bs.

Hooking Scripts can be done in the SmarTeam Script Maintenance that can be opened in **Administrative Tools -> Admin Console -> Script Maintenance**.

1.2. Related Documentation

- Collaborative Systems Engineering – License Use Management (CSE_licenseUseManagement_R19D7.doc)
- Collaborative Systems Engineering – Data Model Documentation (CSE_DataModelDocumentation_R19D7.doc)
- ENOVIA SmarTeam - Editor Installation Guide
- ENOVIA SmarTeam - Editor Administrator Guide
- ENOVIA SmarTeam - Editor User Guide
- ENOVIA SmarTeam - Foundation Installation Guide
- ENOVIA SmarTeam - Foundation Administrator Guide
- ENOVIA SmarTeam - Foundation User Guide

1.3. Definitions

1.3.1. Glossary

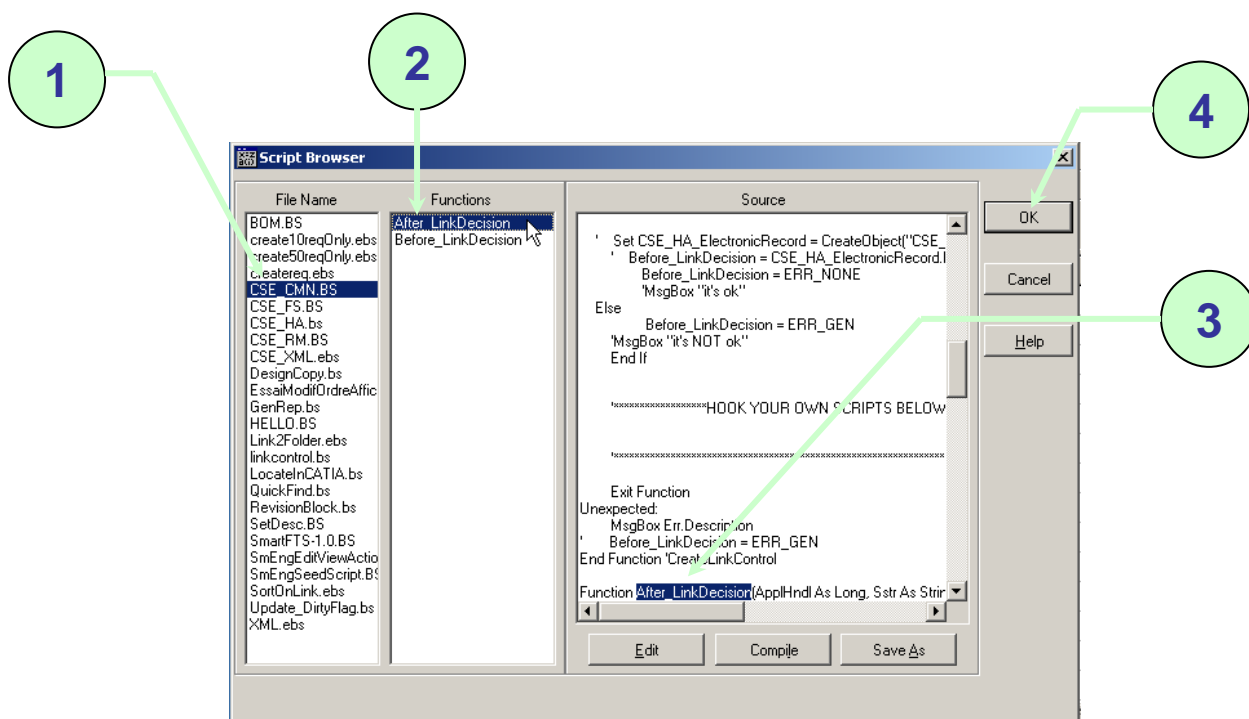
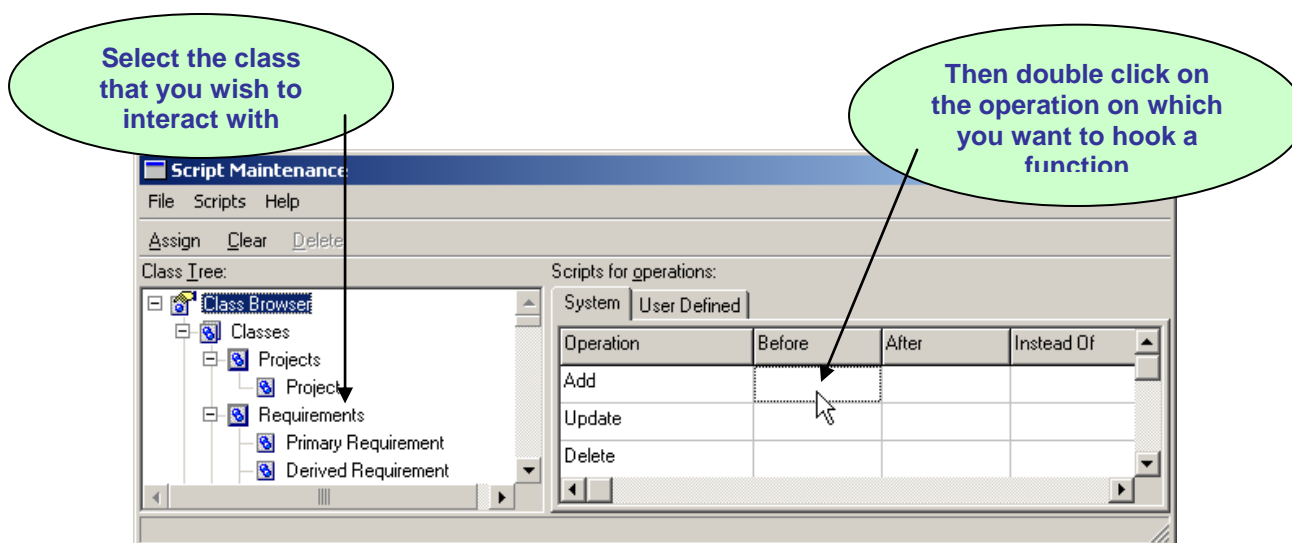
Acronym	Definition
BPA	Business Process Accelerator
CSE	Collaborative System Engineering

Table 1 - Glossary

2. Collaborative Systems Lifecycle Management & Traceability implementation overview

2.1. How to hook script functions

In order to hook a function you can follow the steps described below:



Example : Hooking After_LinkDecision function

- Select the appropriate script on the left. (1)
- Then select the appropriate function. (2)
- The function is highlighted automatically. (3)
- Click on the “OK” button to hook the function chosen. (4)

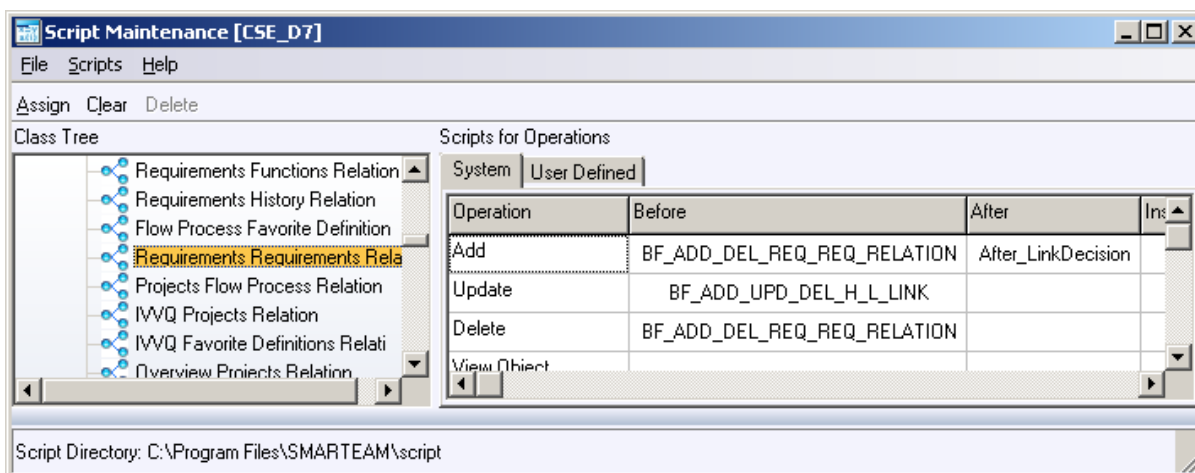
2.2. Link Control only Script Hooking

2.2.1. System tab

There are two different functions that need to be hooked in the SmarTeam Script Maintenance. The script name is **CSE_CMN.bs**.

In case the script needs to be added, SmarTeam scripts directory is generally *C:\Program Files\SMARTTEAM\Script*

In the Script Maintenance, select the type of link on which you want to add a Link Control service. For a normal functioning, you may hook script functions as shown below :



Script functions hooked.

As it is shown, you may hook 3 functions:

- **BF_ADD_DEL_REQ_REQ_RELATION** : Before Add, Before Delete
- **BF_ADD_UPD_DEL_H_L_LINK** : Before Update
- **After_LinkDecision** : After Add

In this example, the "Requirements Requirements Relation" is chosen.

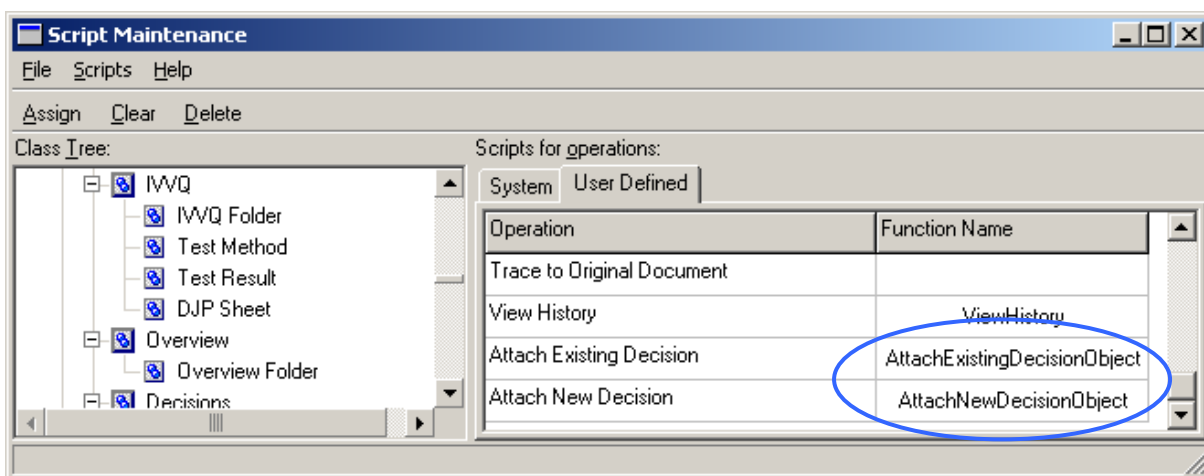
To attach a function to a SmarTeam event, you have to the different steps described in the paragraph 2.1

2.2.2. User Defined tools tab

Then, on the User Defined tools tab, you have to attach the following functions:

- **AttachExistingDecisionObject**: used to attach an existing decision on an object by right clicking on it.
- **AttachNewDecisionObject**: used to create a new decision and attach it to an object by right clicking on it.

The User Defined tools tab looks like shown below:



User defined tools tab for Link control hooks.

NB: If an operation doesn't exist, you just need to write a new line below, giving a name to this new operation, and then you can hook a script in front of it, in the "function name" column.

3. Collaborative Systems Lifecycle Management & Traceability History Management

If you wish to keep trace of operations done when handling links, using the history management services of CSE, you will have to hook the following script functions as shown below.

You can also apply this history management to any objects class, using the same scripts but hooked on the target class.

Nevertheless we do not support that functionality on hierarchical link classes.

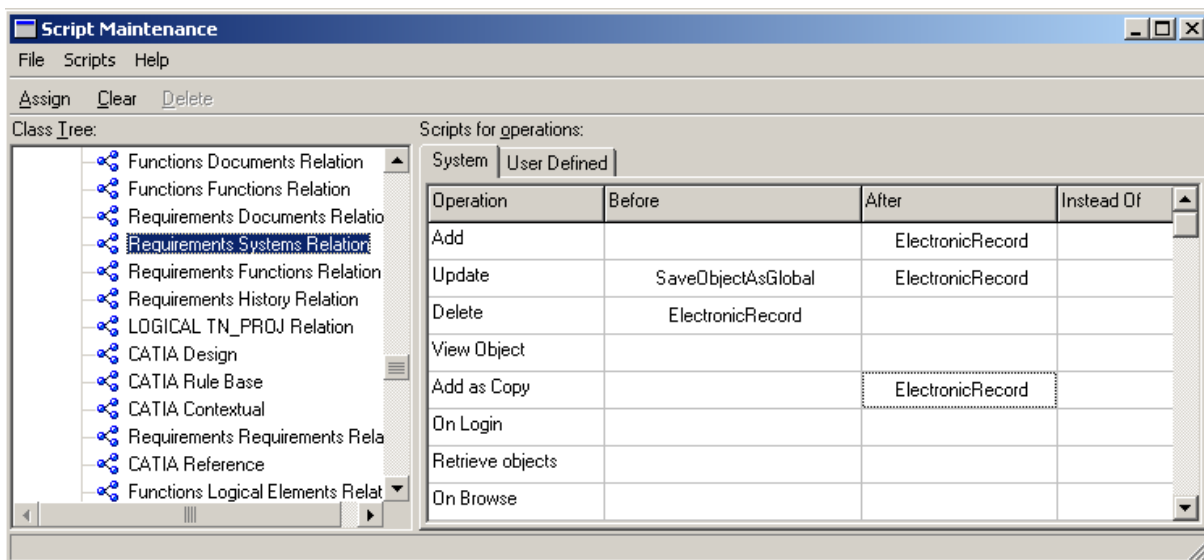
Important notes:

- In order to work properly, it is required that a link class exists between the object class on which the History Management is applied, and the History class. If such a link class does not exist, an error will be displayed when the User Defined Tool is called.
- For some specific classes on which hooks may already exist to support specific functionalities, hooking may differ of the general cases.

The history services script file is named **CSE_HA.bs** for the general case. Nevertheless, when several services are used simultaneously, the script file to use is **Common.bs**.

3.1. System tab

3.1.1. General case

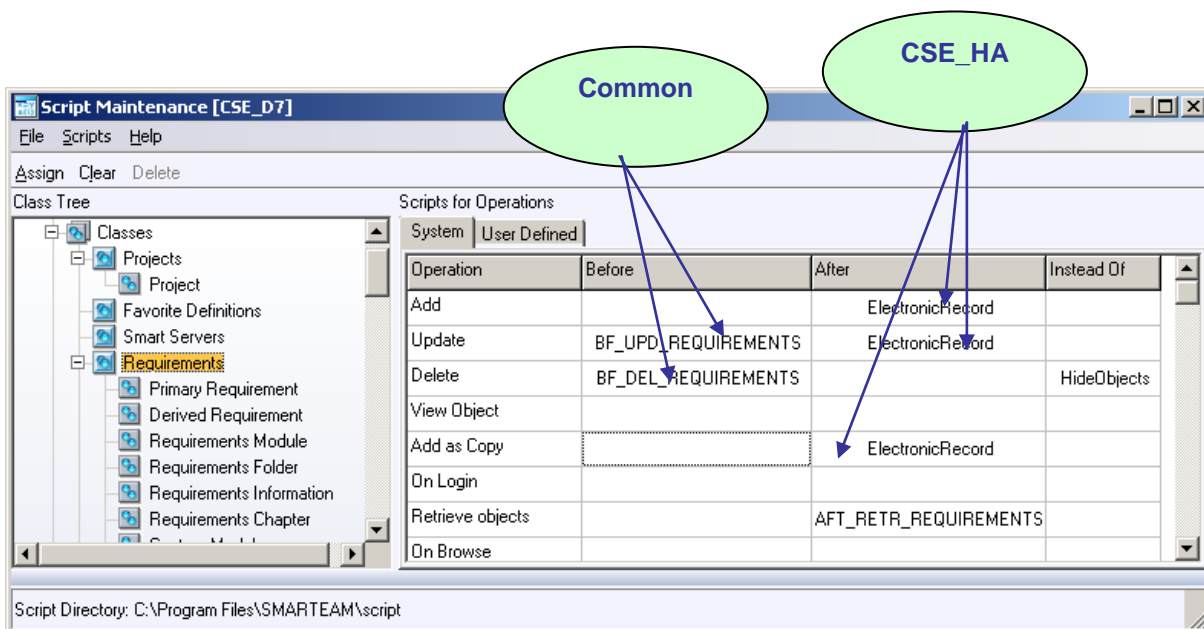


Using History Management: Needed functions for general case

As it is shown, you may hook 2 functions that are in **CSE_HA.bs**:

- **SaveObjectAsGlobal** : Before Update
- **ElectronicRecord** : After Add, After Update, Before Delete, After Add As Copy

3.1.2. Specific case : Requirements class



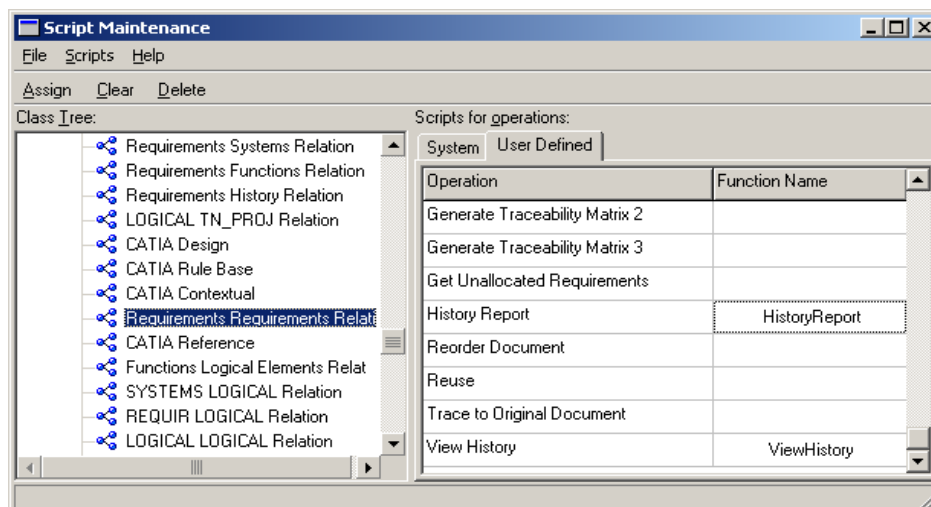
Using History Management: Needed functions for Requirements class

As it is shown, you may hook 3 functions:

- **BF_UPD_REQUIREMENTS** (in **Common.bs**): Before Update
- **BF_DEL_REQUIREMENTS** (in **Common.bs**): Before Delete
- **AFT_RETR_REQUIREMENTS** (in **Common.bs**): After Retrieve Objects
- **ElectronicRecord** (in **CSE_HA.bs**): After Add, After Update, After Add As Copy
- **HideObjects** (in **CSE_HA.bs**): Instead Of Delete

3.2. User Defined tab

Now, on the "User Defined" tab, you may hook the following functions:



History Management: User Defined functions

Still in the CSE_HA.bs script and as it is shown, you may hook 2 functions :

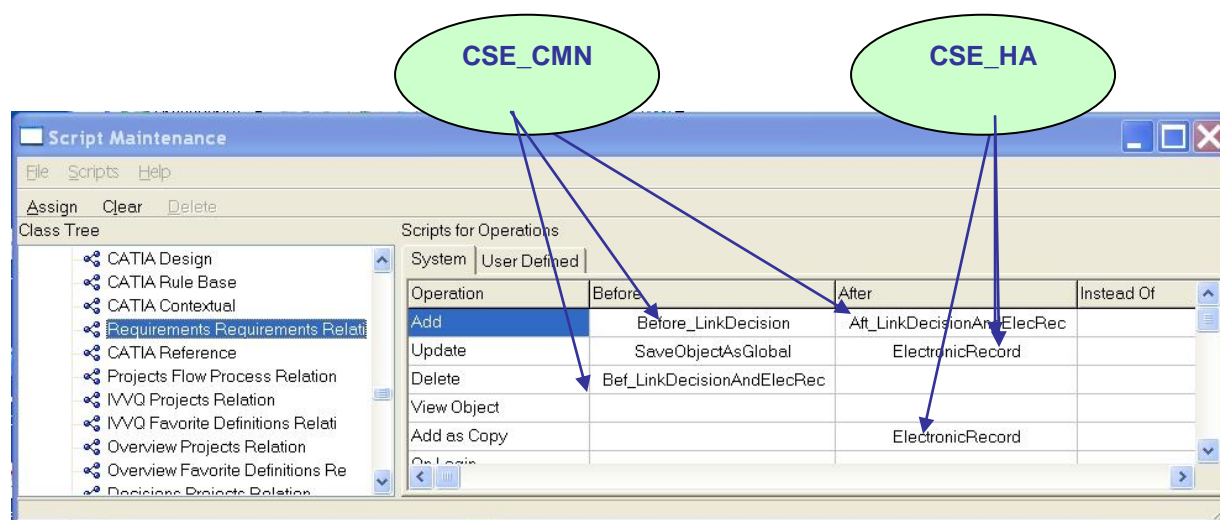
- For the **History Report** : HistoryReport
- For the **View history** : ViewHistory

NB: If an operation doesn't exist, you just need to write a new line below, giving a name to this new operation, and then you can hook a script in front of it, in the "function name" column.

4. Collaborative Systems Lifecycle Management & Traceability

Configuring both History Management and Link Control

Now, if you wish to have both history management and Link Control, you will have to select functions that are in the CSE_CMN.bs and CSE_HA.bs scripts and hook them as shown below :



Configuration for Link Control and History Management

For a normal functioning, you may hook functions the following way:

- **Before_LinkDecision (in CSE_CMN.bs):** Before Add
- **Aft_LinkDecisionAndElecRec (in CSE_CMN.bs):** After Add
- **Bef_LinkDecisionAndElecRec (in CSE_CMN.bs):** Before Delete
- **SaveObjectAsGlobal (in CSE_HA.bs):** Before Update
- **ElectronicRecord (in CSE_HA.bs):** After Update, After Add As Copy

To achieve the configuration describe below, you have to select functions either in CSE_HA.bs or in CSE_CMN.bs script files, as indicated.

NB:

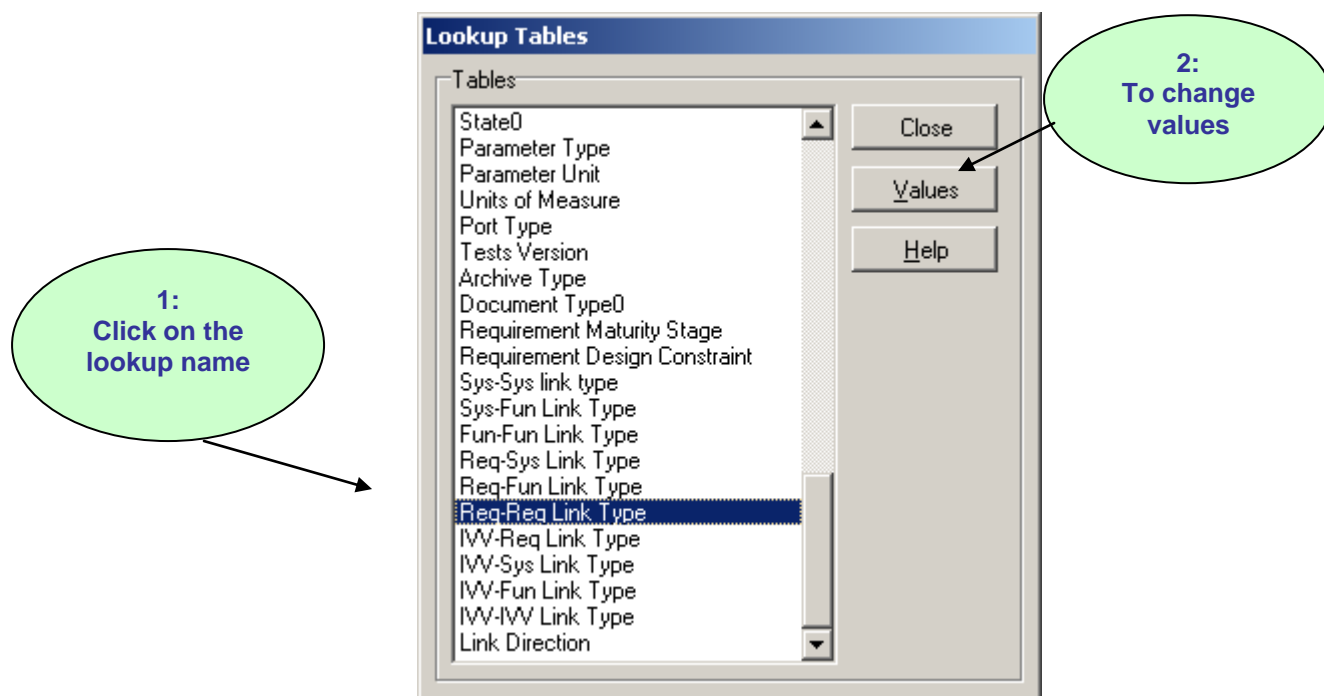
To configure **User Defined tools**, you may follow instructions delivered in chapters 2.2.2 and 3.2.

5. Collaborative Systems Lifecycle Management & Traceability Lookup Tables

This section describes how to fill lookup tables. In the link control window, it is possible to add an attribute about the link type that is going to be created. The choice of the type is made through a SmartTeam lookup table. Here is how to set them up.

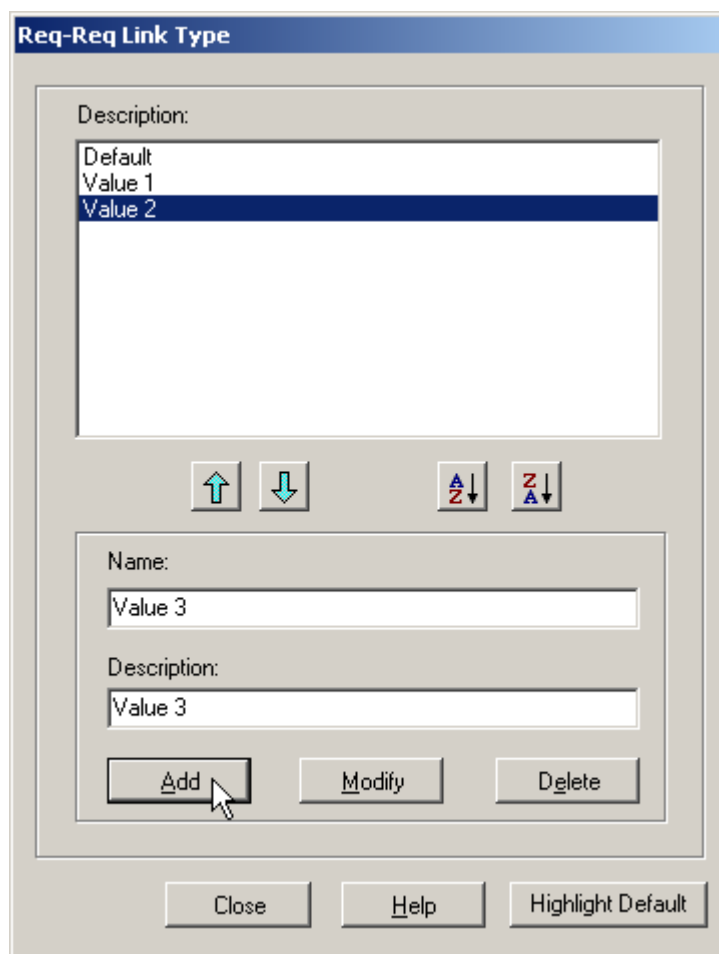
- In SmartTeam, go to the following menu : Tools -> Lookup tables
- Then choose the lookup you want to set.

As an example for this documentation, we will see how to set up the table of the Requirements Requirements link type:



Choosing which lookup table to set up

Now here is how to modify or create new values in the table:



To add a value:

- Fill in the name and description fields. Attention: to avoid confusion, Name and Description fields **must be identical**.
- Click "Add" to add the value

To modify a value:

- Click on a value in the top of the window
- Modify both Name and Description fields
- Ensure Name and Description are identical
- Click on modify

6. Collaborative Systems Lifecycle Management & Traceability Adding Admin Settings for LinkControl

Conventions:

- Rectangular brackets [Conditional Value] specify value that is not known and depends on a condition. The part in blue is a description of the value to be entered, example: [ID of Root Folder] is selected by the user depending on an existing object he wants to specify in the database
- Example** Values are shown in **Bold Orange**, These values can be different according to administrator choice.

The following Admin Settings are necessary in order to automatically place the created or linked decisions objects in a tree structure in SmarTeam. A root Decisions folder object needs to be defined, as well as a name for the sub folders created based on the link or object class:

All admin settings should be placed under a new Admin Folder named "Decision Folders"

Root Folder

Add Admin Setting	
Section	DECISION_FOLDERS
Subject	ROOT_FOLDER
Description	Root Folder
Value1	DECISIONS-0000001 [Unique Identifier of Decision Folder to be used as root]
Value2	
Value3	
Value4	
LongVal	
<Other>	

Decisions Folder Definitions

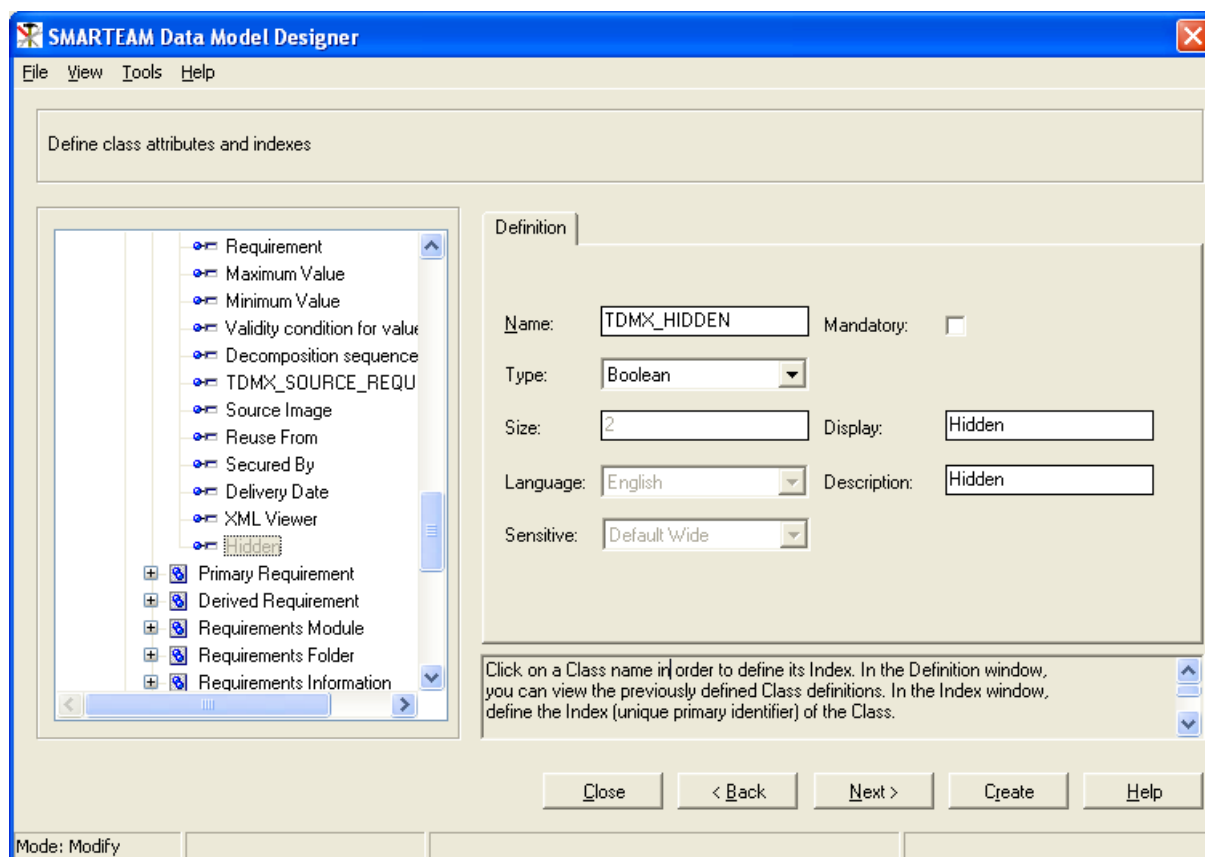
One per each Link Class or Leaf Class to which decisions are attached.

Add Admin Setting	
Section	DECISION_FOLDERS
Subject	REQUIREMENTS_REQUIREMENTS_RELATI [Name of Class or Link Class for which the setting will be valid]
Description	Requirements Requirements Decisions
Value1	Requirements Requirements Decisions [Name of Folder to be created under the Root folder and to which the new or attached decisions shall be linked]
Value2	
Value3	
Value4	
LongVal	
<Other>	

7. Collaborative Systems Lifecycle Management & Traceability Undelete

7.1. Adding attribute

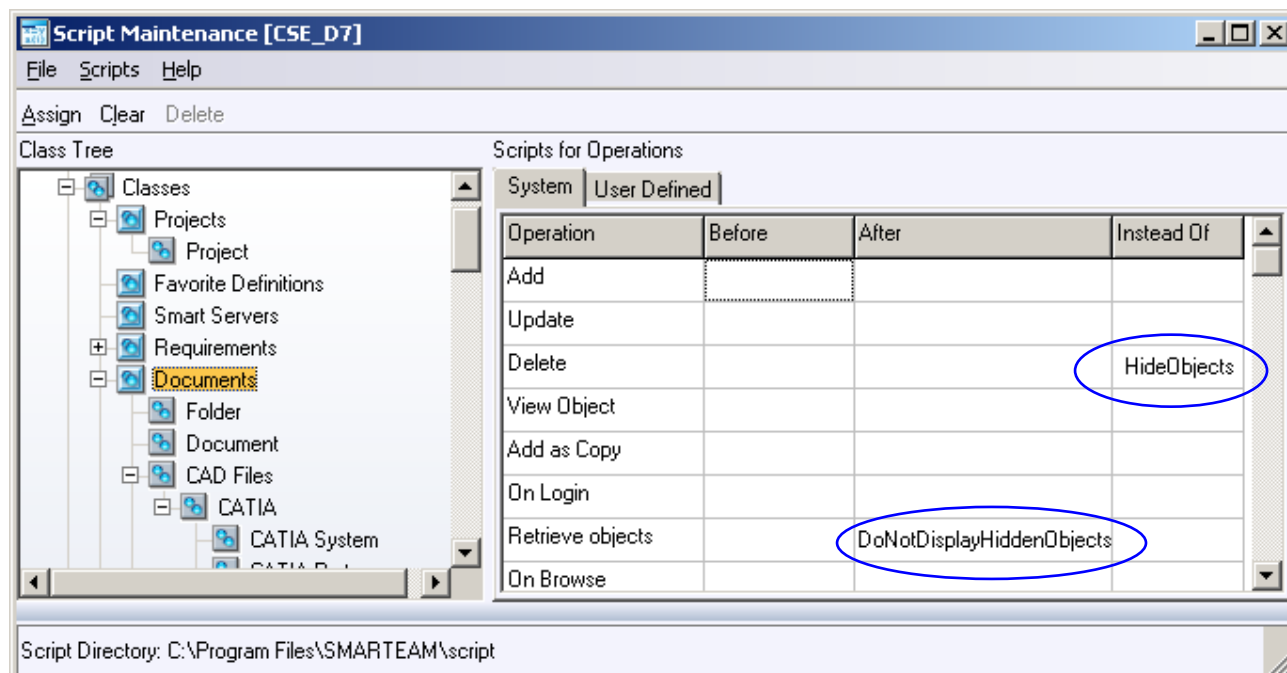
Add the attribute TDMX_HIDDEN (internal name *TDMX_HIDDEN*) on the super classes *Requirements* and *Documents*. This attribute has to be a Boolean.



7.2. Configure "Script Maintenance"

There are four different functions that need to be hooked to events in the *ENOVIA SmarTeam Script Maintenance*. They are described in the following table.

Script Name	Function	Operation	Event
CSE_CMN	HideObjects	Delete	Instead Of
CSE_CMN	DoNotDisplayHiddenObjects	Retrieve objects	After



If a script is already hooked on the event *Retrieve Object*, you can add the following lines at the end of the hooked function in *SmartScript* file.

```
Dim SmSession As SmApplic.SmSession
Dim FirstRec As Object
Dim SecondRec As Object
Dim CSE_CMN As Object

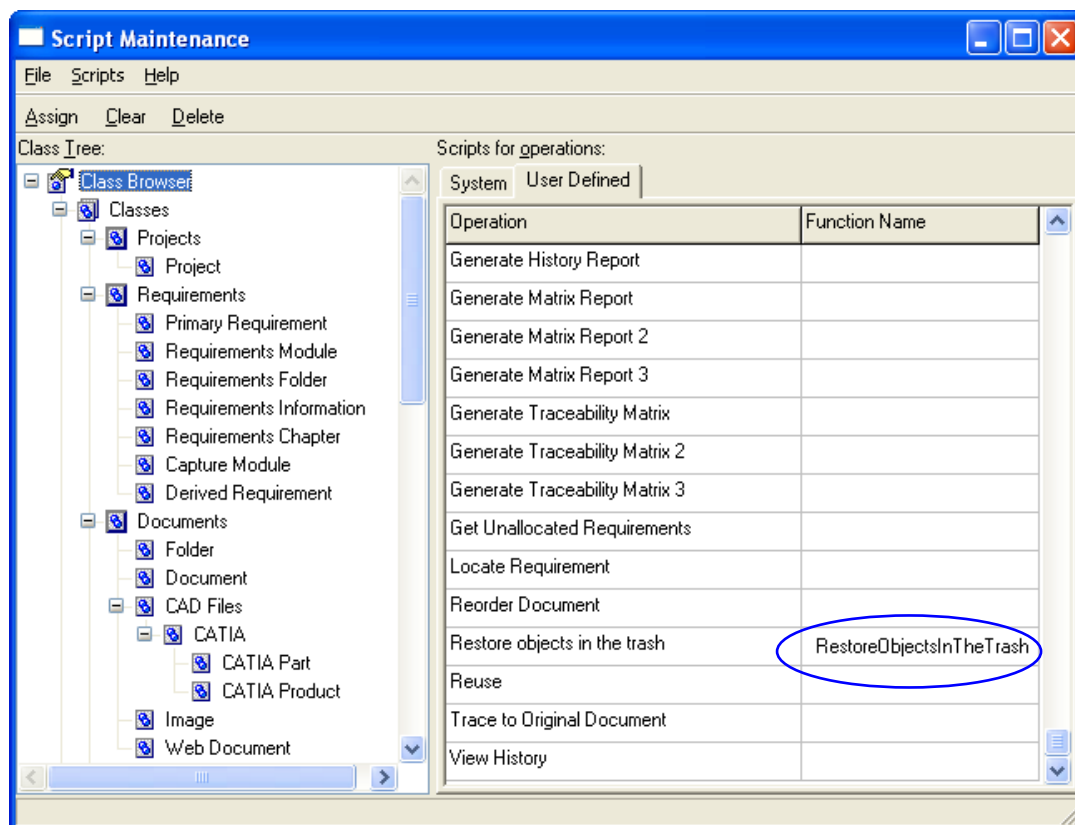
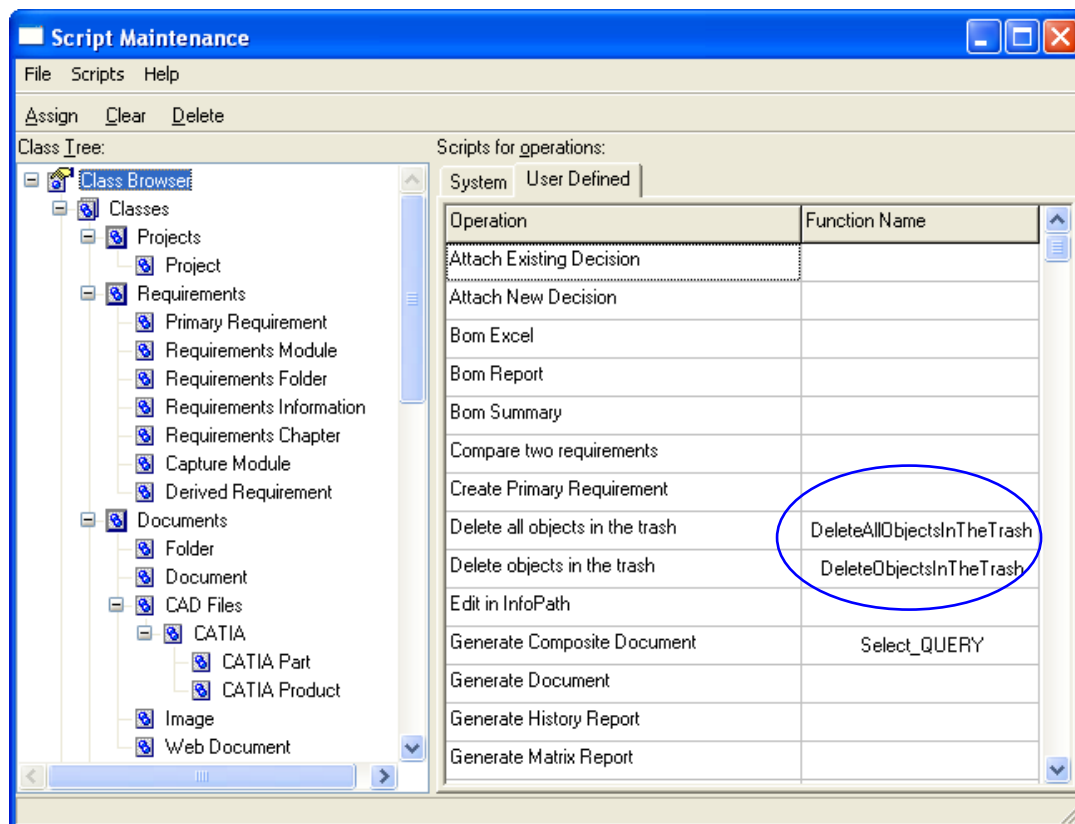
Set SmSession = SCREXT_ObjectForInterface(ApplHndl)
CONV_RecListToComRecordList FirstPar, FirstRec
CONV_RecListToComRecordList SecondPar, SecondRec

Set CSE_CMN = CreateObject("CSE_CMN.Application")
CSE_CMN.DoNotDisplayHiddenObjects SmSession, FirstRec, SecondRec

CONV_ComRecListToRecordList FirstRec, FirstPar
```

There are three different functions that need to be hooked as *User Defined Tools* in the *ENOVIA SmarTeam Script Maintenance*. They are described in the following table.

Script Name	Function	Class
CSE_CMN	DeleteAllObjectsInTheTrash	All classes
CSE_CMN	DeleteObjectsInTheTrash	All classes
CSE_CMN	RestoreObjectsInTheTrash	All classes



8. Collaborative Systems Lifecycle Management & Traceability Verification matrix

8.1. Default verification matrixes

There are three different default verification matrixes:

- Requirement Verification Matrix (RVM)
- Requirement Allocation Matrix (RAM)
- Requirements Compliance Matrix (RCM)

It is important to notice that each verification matrix is identified by a unique identifier used by the generic code to produce the report.

8.1.1. RVM matrix

8.1.1.1. Hook

The RVM matrix generation is hooked on Requirements classes as User defined tool by using the SmartBasic Script Maintenance tool.

The script to hook is defined in CSE_RM.bs file located in <Directory of installation of ENOVIA SmarTeam>/Script.

The script name is GenerateMatrixReport with code:

```
Function GenerateMatrixReport(ApplHndl As Long,Sstr As String,
                             FirstPar As Long,SecondPar As Long,
                             ThirdPar As Long ) As Integer
    Dim SmSession           As SmApplic.SmSession
    Dim FirstRec            As Object
    Dim CSE_RM              As Object

    Set SmSession = SCREXT_ObjectForInterface(ApplHndl)
    SmSession.GlobalData.value("CSE_LIC") = True
    CONV_RecListToComRecordList FirstPar,FirstRec
    Set CSE_RM = CreateObject("CSE_RM.Application")
    CSE_RM.GenerateMatrixReport SmSession, FirstRec
End Function
```

8.1.1.2. Icon in toolbar

The icon is located in :

<Directory of installation of ENOVIA SmarTeam>/Icons.

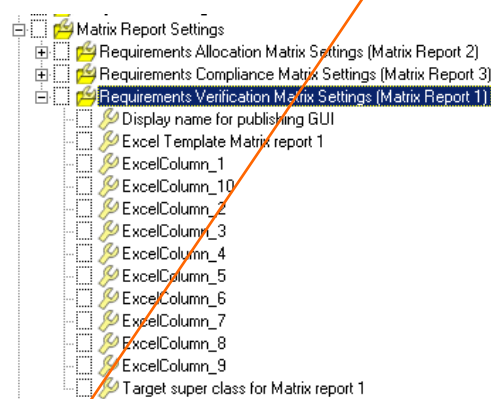
Its name is :

I_NewMatrixRequirement1.bmp

identifier

This icon is built with I_NewMatrixRequirement.bmp image file as template.

8.1.1.3. Admin settings



They are defined under folder **Requirements Management / Matrix Report Settings / Requirements Verification Matrix Settings**. All the admin settings for this matrix use the section **REPORT_MATRIX_1**.

Three subjects are used :

- CLASS
- GUI_DISPLAY_NAME
- EXCEL_COLUMN_x
- EXCEL_TEMPLATE

• CLASS Subject

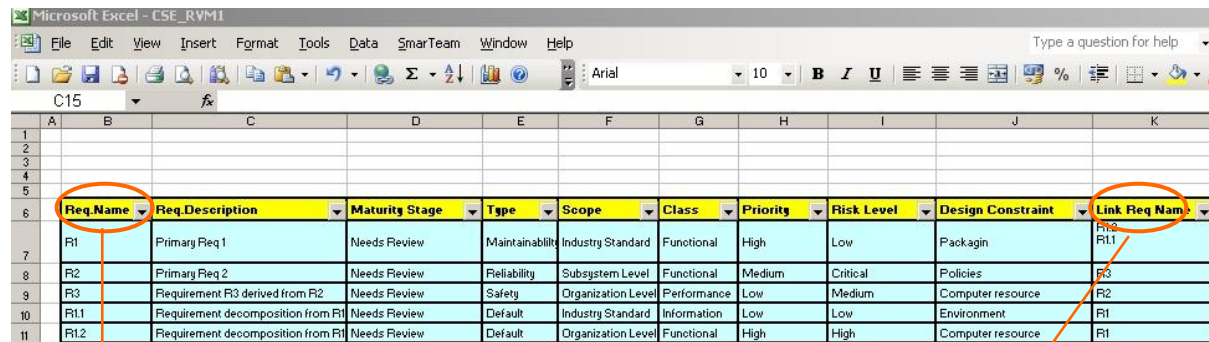
Attribute	Link	Comment
 Admin Setting Business Process Content		
Identifier	ADMIN-000464	
Section	REPORT_MATRIX_1	
Subject	CLASS	
Description	Class	
First Value	Requirements	
Second Value	DRILLDOWN_REQ_2	
Third Value		
Fourth Value		

First value: Name of the super-class used for reporting in the database model
Second value: Name of the Drilldown to use to parse class objects from the user selection in tree class (they are defined under folder **Requirements management / Global Drill down behavior settings**)

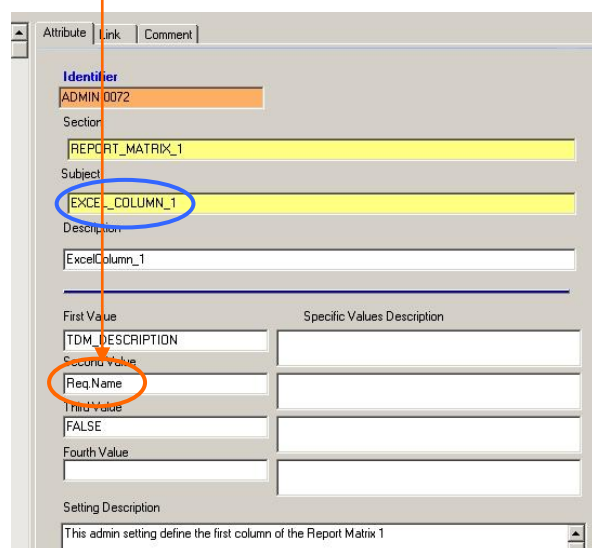
• GUI DISPLAY NAME

First value: Name of the template displayed thru the standard GUI

• EXCEL COLUMN x Subject



Req.Name	Req.Description	Maturity Stage	Type	Scope	Class	Priority	Risk Level	Design Constraint	Link Req Name
R1	Primary Req 1	Needs Review	Maintainability	Industry Standard	Functional	High	Low	Packagin	R1
R2	Primary Req 2	Needs Review	Reliability	Subsystem Level	Functional	Medium	Critical	Policies	R2
R3	Requirement R3 derived from R2	Needs Review	Safety	Organization Level	Performance	Low	Medium	Computer resource	R2
R11	Requirement decomposition from R1	Needs Review	Default	Industry Standard	Information	Low	Low	Environment	R1
R12	Requirement decomposition from R1	Needs Review	Default	Organization Level	Functional	High	High	Computer resource	R1



Attribute | Link | Comment

Identifier
ADMIN 0072

Section
REPORT_MATRIX_1

Subject
EXCEL_COLUMN_1

Description
ExcelColumn_1

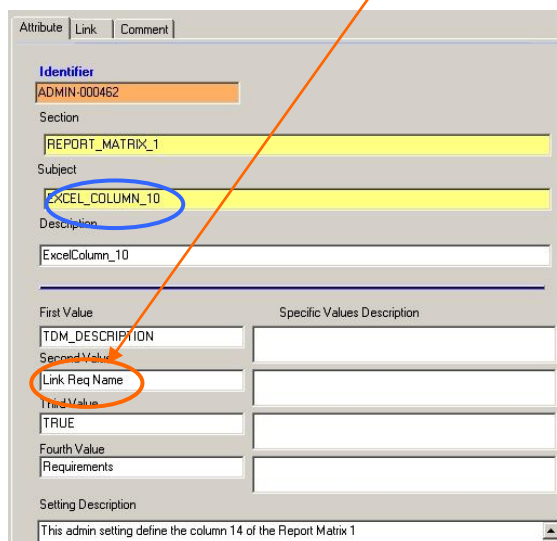
First Value
TDM_DESCRIPTION

Second Value
Req.Name

Third Value
FALSE

Fourth Value

Setting Description
This admin setting define the first column of the Report Matrix 1



Attribute | Link | Comment

Identifier
ADMIN-000462

Section
REPORT_MATRIX_1

Subject
EXCEL_COLUMN_10

Description
ExcelColumn_10

First Value
TDM_DESCRIPTION

Second Value
Link Req Name


Third Value
TRUE

Fourth Value
Requirements

Setting Description
This admin setting define the column 14 of the Report Matrix 1

First value: Attribute name of class (or linked class if third value is true) in the data model
Second value: Caption of attribute to display in matrix
Third value: False if the attribute belongs to class or True if attribute belongs to a linked class

• EXCEL TEMPLATE Subject

Attribute	Link	Comment
 Admin Setting Business Process Content		
Identifier		
ADMIN-0145		
Section		
REPORT_MATRIX_1		
Subject		
EXCEL_TEMPLATE		
Description		
Excel Template Matrix report 1		
First Value	Specific Values	
CSE_RVM.xlt		
Second Value		
Document		
Third Value		
Microsoft Excel		
Fourth Value		

- First value:** Name of file used as template (the path to access the file is given by the admin setting defined by subject `RELATIVE_DIRECTORY` under `TEMPLATE_PATH` section)
- Second value:** Type of file
- Third value:** Tool or software that generates the template

• RVM Admin Settings default values

Note

See Requirements Management implementation guide to get the default admin settings for the RVM matrix.

8.1.2. RAM matrix

8.1.2.1. Hook

The RAM matrix generation is hooked on Requirements classes as User defined tool by using the SmartBasic Script Maintenance tool.

The script to hook is defined in CSE_RM.bs file located in <Directory of installation of ENOVIA SmarTeam>/Script.

The script name is GenerateMatrixReport with code:

```
Function GenerateMatrixReport2(ApplHndl As Long, Sstr As String,
                             FirstPar As Long, SecondPar As Long,
                             ThirdPar As Long ) As Integer
    Dim SmSession           As SmApplic.SmSession
    Dim FirstRec            As Object
    Dim CSE_RM              As Object

    Set SmSession = SCREXT_ObjectForInterface(ApplHndl)
    SmSession.GlobalData.value("CSE_LIC") = True
    CONV_RecListToComRecordList FirstPar, FirstRec
    Set CSE_RM = CreateObject("CSE_RM.Application")
    CSE_RM.GenerateMatrixReport SmSession, FirstRec
End Function
```

8.1.2.2. Icon in toolbar

The icon is located in :

<Directory of installation of ENOVIA SmarTeam>/Icons.

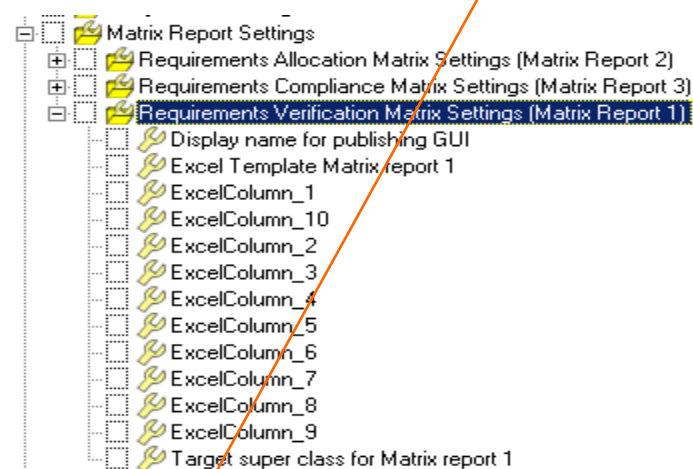
Its name is :

I_NewMatrixRequirement2.bmp

This icon is built with I_NewMatrixRequirement.bmp image file as template.

identifier

8.1.2.3. Admin settings



They are defined under folder **Collaborative Services / Matrix Report Settings / Requirements Allocation Matrix Settings**. All the admin settings for this matrix use the section **REPORT_MATRIX_2**.

As for RVM matrix, three subjects with identical signification are used for RAM admin settings:

- CLASS
- GUI_DISPLAY_NAME
- EXCEL_COLUMN_x
- EXCEL_TEMPLATE

• **RAM Admin Settings default values**

Note

See Requirements Management implementation guide to get the default admin settings for the RAM matrix.

8.1.3. RCM matrix

8.1.3.1. Hook

The RCM matrix generation is hooked on Requirements classes as User defined tool by using the SmartBasic Script Maintenance tool.

The script to hook is defined in CSE_RM.bs file located in <Directory of installation of ENOVIA SmarTeam>/Script.

The script name is GenerateMatrixReport with code:

```
Function GenerateMatrixReport3(ApplHndI As Long, Sstr As String,
                              FirstPar As Long, SecondPar As Long,
                              ThirdPar As Long ) As Integer
    Dim SmSession           As SmApplic.SmSession
    Dim FirstRec            As Object
    Dim CSE_RM              As Object

    Set SmSession = SCREXT_ObjectForInterface(ApplHndI)
    SmSession.GlobalData.value("CSE_LIC") = True
    CONV_RecListToComRecordList FirstPar, FirstRec
    Set CSE_RM = CreateObject("CSE_RM.Application")
    CSE_RM.GenerateMatrixReport SmSession, FirstRec
End Function
```

8.1.3.2. Icon in toolbar

The icon is located in :

<Directory of installation of ENOVIA SmarTeam>/Icons.

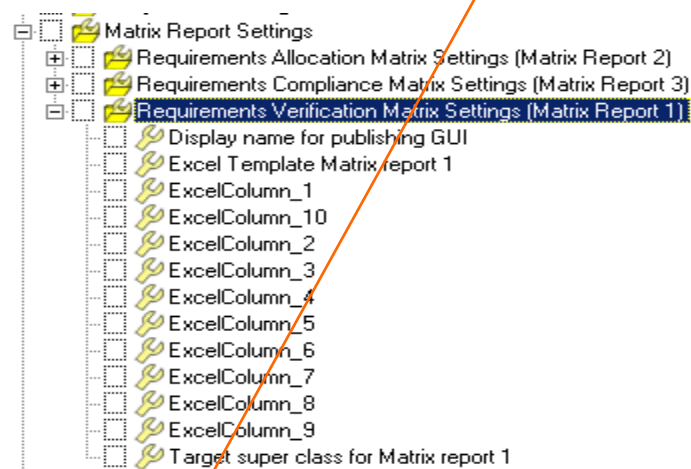
Its name is :

I_NewMatrixRequirement3.bmp

This icon is built with I_NewMatrixRequirement.bmp image file as template.

identifier

8.1.3.3. Admin settings



They are defined under folder **Collaborative Services / Matrix Report Settings / Requirements Compliance Matrix Settings**. All the admin settings for this matrix use the section **REPORT_MATRIX_3**.

As for RVM matrix, three subjects with identical signification are used for RCM admin settings:

- CLASS
- GUI_DISPLAY_NAME
- EXCEL_COLUMN_x
- EXCEL_TEMPLATE

• **RCM Admin Settings default values**

Note

See Requirements Management implementation guide to get the default admin settings for the RCM matrix.

8.2. Adding user customized verification matrix

- 1) It is first necessary to attribute a unique identifier to the new verification matrix (not used by the others existing verification matrixes to avoid conflict)
- 2) The administrator shall add an admin settings section with name REPORT_MATRIX_<unique identifier> with all subjects necessary for verification matrix :
 - CLASS
 - GUI_DISPLAY_NAME
 - EXCEL_COLUMN_x
 - EXCEL_TEMPLATE

9. Collaborative Systems Lifecycle Management & Traceability

Traceability matrix

9.1. *Default traceability matrixes*

There are three different default traceability matrixes:

- Requirements to Requirements traceability matrix
- Requirements to Functions traceability matrix
- Functions to Systems traceability matrix
- Requirements to IVVQ traceability matrix

9.1.1. Requirements to Requirements traceability matrix (unique identifier = 1)

9.1.1.1. Hook

The Requirements to Requirements traceability matrix generation is hooked on Projects classes as User defined tool by using the SmartBasic Script Maintenance tool.

The script to hook is defined in CSE_RM.bs file located in <Directory of installation of ENOVIA SmarTeam>/Script.

The script name is GenerateTraceabilityReport with code:

```
Function GenerateTraceabilityMatrix(ApplHndl As Long, Sstr As String,
                                   FirstPar As Long, SecondPar As Long,
                                   ThirdPar As Long ) As Integer

    Dim SmSession           As SmApplic.SmSession
    Dim FirstRec            As Object
    Dim CSE_RM              As Object

    Set SmSession = SCREXT_ObjectForInterface(ApplHndl)
    SmSession.GlobalData.value("CSE_LIC") = True
    CONV_RecListToComRecordList FirstPar, FirstRec
    Set CSE_RM = CreateObject("CSE_RM.Application")
    CSE_RM.GenerateTraceabilityMatrix SmSession, FirstRe
End Function
```

9.1.1.2. Icon in toolbar

The icon is located in :

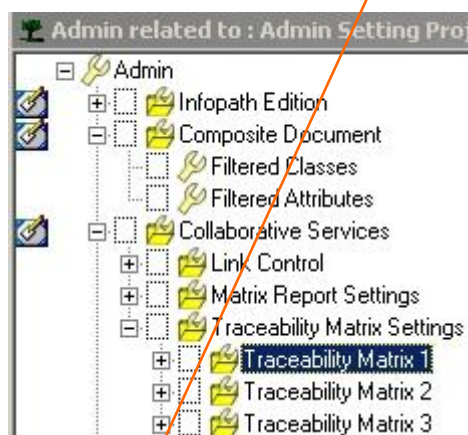
<Directory of installation of ENOVIA SmarTeam>/Icons.

Its name is :

I_TraceabilityMatrix1.bmp

identifier

9.1.1.3. Admin settings

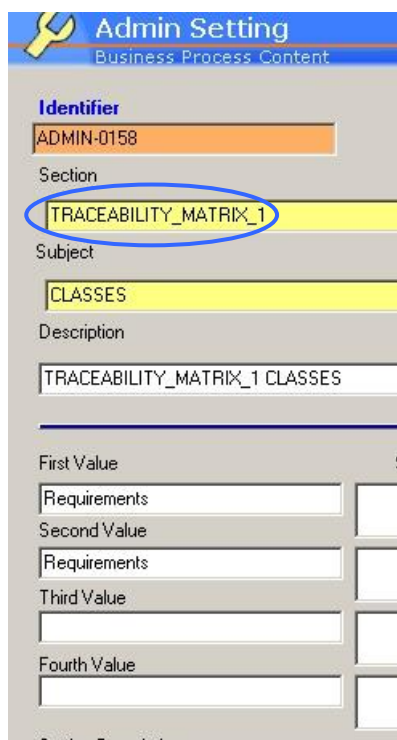


They are defined under folder **Collaborative Services / Traceability Matrix Settings / Traceability Matrix 1**. All the admin settings for this matrix use the section **TRACEABILITY_MATRIX_1**.

Nine subjects are used :

- CLASSES
- GUI_DISPLAY_NAME
- DEFINITION
- DISPLAY_CLASS_ATTRIBUTE
- DISPLAY_LINK_ATTRIBUTES
- DRILLDOWN_BEHAVIORS
- FILTER_LINK_ATTRIBUTE
- LINK_COLORED_ATTRIBUTE
- SELECTION_BEHAVIOR
- EXCEL_TEMPLATE

• CLASSES Subject



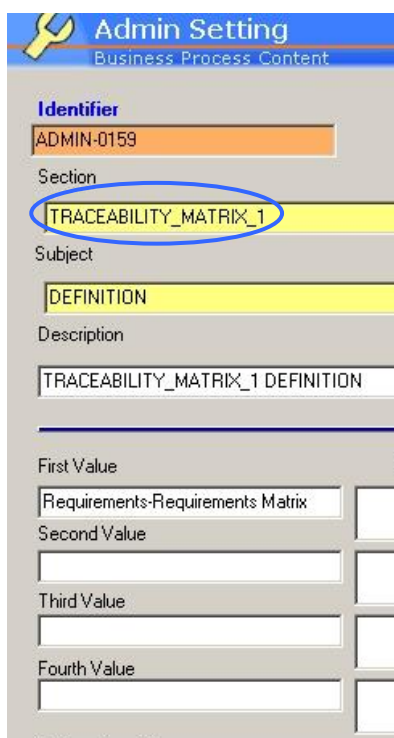
Gives the attributes super-classes used for rows and columns.

First value: Name of the super-class used in columns
Second value: Name of the super-class used in rows

• GUI DISPLAY NAME

First value: Name of the template displayed thru the standard GUI

• DEFINITION Subject

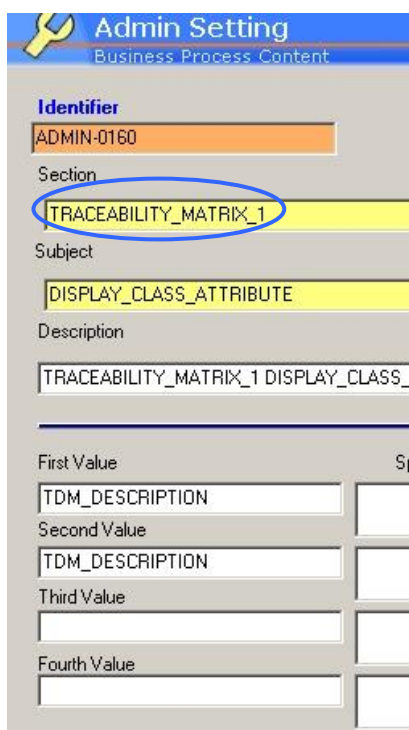


Admin Setting	
Business Process Content	
Identifier	ADMIN-0159
Section	TRACEABILITY_MATRIX_1
Subject	DEFINITION
Description	TRACEABILITY_MATRIX_1 DEFINITION
First Value	Requirements-Requirements Matrix
Second Value	
Third Value	
Fourth Value	

Gives the title of the matrix.

First value: Title of the matrix

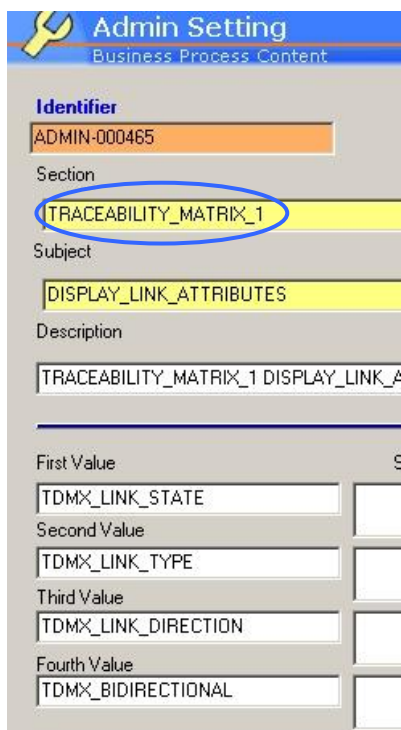
• DISPLAY CLASS ATTRIBUTE Subject



Gives the attributes of row and column class to display as row and column header.

First value: Name of attribute to display for super-class in columns
Second value: Name of attribute to display for super-class in rows

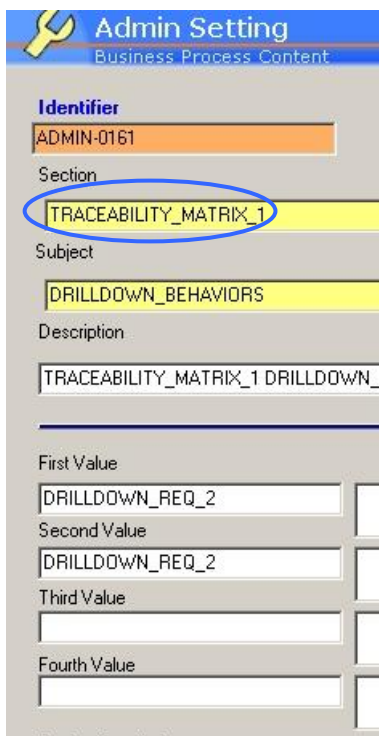
• DISPLAY LINK ATTRIBUTE Subject



Gives the attributes of linked class objects to display in cells (if none a cross is displayed, a maximum of 4 attributes is authorized)

First value: Name of first attribute of link class to display in cell
Second value: Name of second attribute of link class to display in cell
Third value: Name of third attribute of link class to display in cell
Fourth value: Name of fourth attribute of link class to display in cell

• **DRILLDOWN BEHAVIORS Subject**



Admin Setting
Business Process Content

Identifier
ADMIN-0161

Section
TRACEABILITY_MATRIX_1

Subject
DRILLDOWN_BEHAVIORS

Description
TRACEABILITY_MATRIX_1 DRILLDOWN_

First Value
DRILLDOWN_REQ_2

Second Value
DRILLDOWN_REQ_2

Third Value

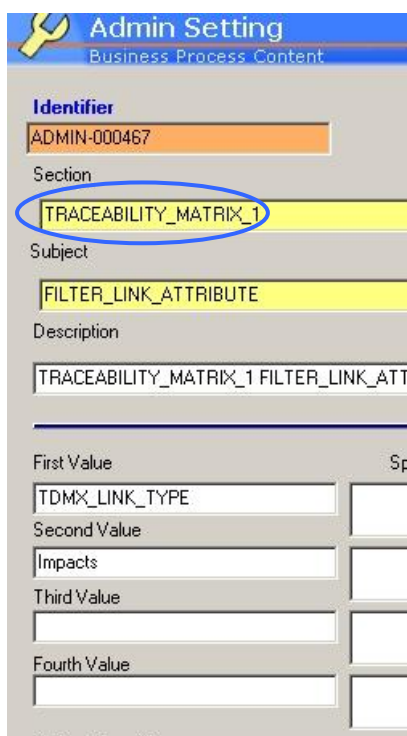
Fourth Value

Gives the parsing mode of tree class objects (row and column).

First value: Name of the drill down used for class in columns

Second value: Name of the drill down used for class in rows

• **FILTER LINK ATTRIBUTE Subject**



Admin Setting
Business Process Content

Identifier
ADMIN-000467

Section
TRACEABILITY_MATRIX_1

Subject
FILTER_LINK_ATTRIBUTE

Description
TRACEABILITY_MATRIX_1 FILTER_LINK_ATT

First Value
TDMX_LINK_TYPE

Second Value
Impacts

Third Value

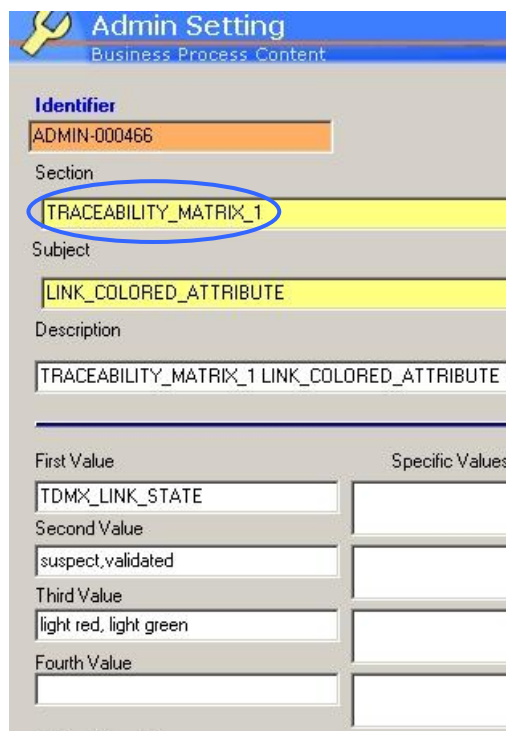
Fourth Value

Gives a condition to be verified for an attribute of link class.

First value: Name of the link class attribute to filter

Second value: Condition to verify for this attribute

• LINK COLORED ATTRIBUTE Subject

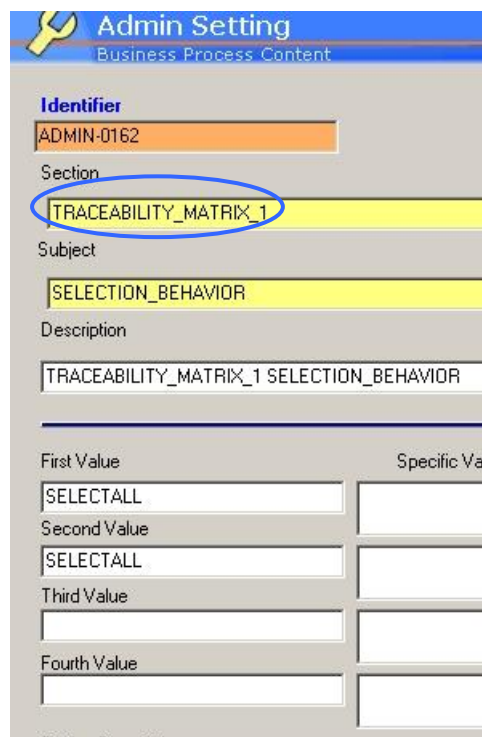


First Value	Specific Values
TDMX_LINK_STATE	
suspect, validated	
light red, light green	

Gives the attribute to be displayed colored.

First value: Name of the link class attribute that will appear colored
Second value: Possible values of this link class attribute
Third value: Corresponding color for each possible value of attribute

• SELECTION BEHAVIOR Subject



Admin Setting
Business Process Content

Identifier
ADMIN-0162

Section
TRACEABILITY_MATRIX_1

Subject
SELECTION_BEHAVIOR

Description
TRACEABILITY_MATRIX_1 SELECTION_BEHAVIOR

First Value **Specific Value**
SELECTALL

Second Value
SELECTALL

Third Value

Fourth Value

Gives the selection mode of row and column class objects.

First value:


Selection mode used for column class objects

- MANUAL for manual selection in column class tree
- SELECTALL for selection of all column class objects linked to current project
- SELECTPROJECTMANUAL for manual selection of a project and manual selection in column class tree
- SELECTPROJECTALL for manual selection of a project and selection of all column class objects linked to the selected project

Second value: Selection mode used for row class objects

- MANUAL for manual selection in row class tree
- SELECTALL for selection of all row class objects linked to current project
- SELECTPROJECTMANUAL for manual selection of a project and manual selection in row class tree
- SELECTPROJECTALL for manual selection of a project and selection of all row class objects linked to the selected project

- EXCEL TEMPLATE Subject



Gives the Excel template name used for reporting. The path to access this template is given by the admin settings with subject `RELATIVE_DIRECTORY` under `TEMPLATE_PATH` section.

First value: Name of file used as template
Second value: Type of file
Third value: Tool or software that generates the template

- Requirements-Requirements traceability matrix Admin Settings default values

Note

See Requirements Management implementation guide to get the default admin settings for the Requirements-Requirement traceability matrix.

9.1.2. Requirements to Functions traceability matrix (unique identifier = 2)

9.1.2.1. Hook

The Requirements to Functions traceability matrix generation is hooked on Projects classes as User defined tool by using the SmartBasic Script Maintenance tool.

The script to hook is defined in CSE_RM.bs file located in <Directory of installation of ENOVIA SmarTeam>/Script.

The script name is GenerateTraceabilityMatrix with code:

```
Function GenerateTraceabilityMatrix2(ApplHndl As Long, Sstr As String,
                                     FirstPar As Long, SecondPar As Long,
                                     ThirdPar As Long ) As Integer
    Dim SmSession      As SmApplic.SmSession
    Dim FirstRec       As Object
    Dim CSE_RM         As Object

    Set SmSession = SCREXT_ObjectForInterface(ApplHndl)
    SmSession.GlobalData.value("CSE_LIC") = True
    CONV_ReclListToComRecordList FirstPar, FirstRec
    Set CSE_RM = CreateObject("CSE_RM.Application")
    CSE_RM.GenerateTraceabilityMatrix SmSession, FirstRec
End Function
```

9.1.2.2. Icon in toolbar

The icon is located in :

<Directory of installation of ENOVIA SmarTeam>/Icons.

Its name is :

I_TraceabilityMatrix1.bmp

identifier

9.1.2.3. Admin settings

They are defined under folder **Collaborative Services / Traceability Matrix Settings / Traceability Matrix**

2. All the admin settings for this matrix use the section TRACEABILITY_MATRIX_2.

As for Requirements-Requirements traceability matrix, nine subjects with identical signification are used for Requirements-Functions traceability matrix admin settings:

- CLASSES
- GUI_DISPLAY_NAME
- DEFINITION
- DISPLAY_CLASS_ATTRIBUTE
- DISPLAY_LINK_ATTRIBUTES
- DRILLDOWN_BEHAVIORS
- FILTER_LINK_ATTRIBUTE
- LINK_COLORED_ATTRIBUTE
- SELECTION_BEHAVIOR
- EXCEL_TEMPLATE

• Requirements to Functions traceability matrix Admin Settings default values

Note

See Requirements Management implementation guide to get the default admin settings for the Requirements to Functions traceability matrix.

9.1.3. Functions to Systems traceability matrix (unique identifier = 3)

9.1.3.1. Hook

The Functions to Systems traceability matrix generation is hooked on Projects classes as User defined tool by using the SmartBasic Script Maintenance tool.

The script to hook is defined in CSE_RM.bs file located in <Directory of installation of ENOVIA SmarTeam>/Script.

The script name is GenerateTraceabilityMatrix with code:

```
Function GenerateTraceabilityMatrix3(ApplHndl As Long, Sstr As String,
                                     FirstPar As Long, SecondPar As Long,
                                     ThirdPar As Long ) As Integer
    Dim SmSession           As SmApplic.SmSession
    Dim FirstRec            As Object
    Dim CSE_RM              As Object

    Set SmSession = SCREXT_ObjectForInterface(ApplHndl)
    SmSession.GlobalData.value("CSE_LIC") = True
    CONV_RecListToComRecordList FirstPar, FirstRec
    Set CSE_RM = CreateObject("CSE_RM.Application")
    CSE_RM.GenerateTraceabilityMatrix SmSession, FirstRec
End Function
```

9.1.3.2. Icon in toolbar

The icon is located in :

<Directory of installation of ENOVIA SmarTeam>/Icons.

Its name is :

I_TraceabilityMatrix1.bmp

9.1.3.3. Admin settings

They are defined under folder **Collaborative Services / Traceability Matrix Settings / Traceability Matrix**

3. All the admin settings for this matrix use the section TRACEABILITY_MATRIX_3

As for Requirements-Requirements traceability matrix, nine subjects with identical signification are used for Functions-Systems traceability matrix admin settings:

- CLASSES
- GUI_DISPLAY_NAME
- DEFINITION
- DISPLAY_CLASS_ATTRIBUTE
- DISPLAY_LINK_ATTRIBUTES
- DRILLDOWN_BEHAVIORS

- FILTER_LINK_ATTRIBUTE
- LINK_COLORED_ATTRIBUTE
- SELECTION_BEHAVIOR
- EXCEL_TEMPLATE

• Functions to Systems Admin Settings default values

Note

See Requirements Management implementation guide to get the default admin settings for the Functions to Systems traceability matrix.

9.1.4. Requirements to IVVQ traceability matrix (unique identifier = 4)

9.1.4.1. Hook

The Functions to Systems traceability matrix generation is hooked on Projects classes as User defined tool by using the SmartBasic Script Maintenance tool.

The script to hook is defined in CSE_RM.bs file located in <Directory of installation of ENOVIA SmarTeam>/Script.

The script name is GenerateTraceabilityMatrix with code:

```
Function GenerateTraceabilityMatrix3(ApplHndl As Long,Sstr As String,
                                   FirstPar As Long,SecondPar As Long,
                                   ThirdPar As Long ) As Integer
    Dim SmSession           As SmApplic.SmSession
    Dim FirstRec            As Object
    Dim CSE_RM              As Object

    Set SmSession = SCREXT_ObjectForInterface(ApplHndl)
    SmSession.GlobalData.value("CSE_LIC") = True
    CONV_RecListToComRecordList FirstPar,FirstRec
    Set CSE_RM = CreateObject("CSE_RM.Application")
    CSE_RM.GenerateTraceabilityMatrix SmSession, FirstRec
End Function
```

9.1.4.2. Icon in toolbar

The icon is located in :

<Directory of installation of ENOVIA SmarTeam>/Icons.

Its name is :

I_TraceabilityMatrix1.bmp

identifier

9.1.4.3. Admin settings

They are defined under folder **Collaborative Services / Traceability Matrix Settings / Traceability Matrix**

3. All the admin settings for this matrix use the section TRACEABILITY_MATRIX_4.

As for Requirements-Requirements traceability matrix, nine subjects with identical signification are used for Functions-Systems traceability matrix admin settings:

- CLASSES
- GUI_DISPLAY_NAME
- DEFINITION
- DISPLAY_CLASS_ATTRIBUTE
- DISPLAY_LINK_ATTRIBUTES
- DRILLDOWN_BEHAVIORS
- FILTER_LINK_ATTRIBUTE
- LINK_COLORED_ATTRIBUTE
- SELECTION_BEHAVIOR
- EXCEL_TEMPLATE

• **Requirements to IVVQ Admin Settings default values**

Note

See Requirements Management implementation guide to get the default admin settings for the Requirements to IVVQ traceability matrix.

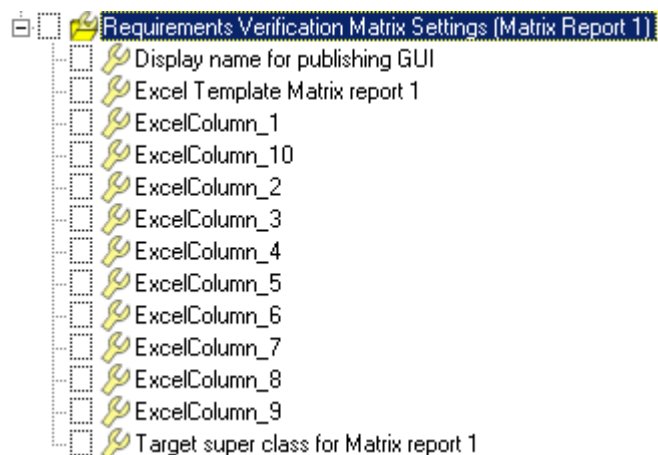
9.2. Adding user customized traceability matrix

- 1) It is first necessary to attribute a unique identifier to the new traceability matrix (not used by the others existing verification matrixes to avoid conflict)
- 2) The administrator shall add an admin settings section with name TRACEABILITY_MATRIX_<unique identifier> with all subjects necessary for traceability matrix :
 - CLASSES
 - GUI_DISPLAY_NAME
 - EXCEL_TEMPLATE
 - DEFINITION
 - DISPLAY_CLASS_ATTRIBUTE
 - DISPLAY_LINK_ATTRIBUTES
 - DRILLDOWN_BEHAVIORS
 - SELECTION_BEHAVIOR
 - FILTER_LINK_ATTRIBUTE
 - LINK_COLORED_ATTRIBUTE

10. Appendix A – Admin Settings rules

10.1. Settings Folder

So as to group a set of parameters, you can use the *Settings Folder*. The *Settings Folder* class is associated to the attribute *Section* (if the corresponding field is not present on the *profile card*, add it). All the *Admin Settings* whose the *Section* is the same have to be hierarchically linked to a unique *Settings Folder*. And the *Section* of this *Folder* will also be the same than the *Admin Settings* below it.



It is especially important for the functions *Document Report*, *Matrix Report* and *Traceability Matrix* which suppose the choice of a set of parameters.

10.2. Avoid doubles

Each *Admin Setting* has to be identified by the *Section* and the *Subject*. The script *CSECMN_TRG_SCRIPT_01* is in charge to respect this rule.

Script Name	Function	Operation	Event	Class
CSE_CMN	CSECMN_TRG_SCRIPT_01	Add	Before	Admin Setting (TN_ADMIN_SETTING)
CSE_CMN	CSECMN_TRG_SCRIPT_01	Update	Before	Admin Setting (TN_ADMIN_SETTING)
CSE_CMN	CSECMN_TRG_SCRIPT_01	Add as Copy	Before	Admin Setting (TN_ADMIN_SETTING)

But in the current database, there are maybe doubles of *Admin Settings* which can cause problems during the execution of most of scripts. So as to remove all the doubles, use the script *CSECMN_UDT_SCRIPT_02*. The *CSECMN_UDT_SCRIPT_02* is a UDT and the “User Define” name is “Check Admin Settings Doubles”

Script Name	Function	Class
CSE_CMN	CSECMN_UDT_SCRIPT_02	Projects (TN_PROJECTS)

10.3. Remove the unused Admin Settings

It is possible that there are unused *Admin Settings*. The most often these *Admin Settings* are not present in the *Admin Tree*. The script *CSECMN_UDT_SCRIPT_01* is in charge to get back all the *Admin Settings* which are not associated to the *Admin Tree*.

The *CSECMN_UDT_SCRIPT_01* is a UDT and the “User Define” name is “Get back free Admin Settings”

Script Name	Function	Class
<i>CSE_CMN</i>	<i>CSECMN_UDT_SCRIPT_01</i>	Projects (TN_PROJECTS)