

# Requirements XML Edition

- RX9 -



## **Implementation Guide – V1**

**BPA Delivery 7 for V5R19 (V5.7)**

# Modification Tracking

Version	Date	Done by	Modification
	07 Oct 2008	CBZ	Document creation
D6W42.3	15 Oct 2008	NJZ/CBZ	Document verification
D6W48.3	25 Nov 2008	CBZ	Special characters are allowed in Infopath and Profile card of IVVQ and Requirements objects. Add XHTML attributes for IVVQ objects
D6W5.3	28 Jan 2009	CBZ	Add TDMX_LOCKED_BY attribute for IVVQ objects
D7W22.3	14 May 2009	CBZ	Add IVVQ Analysis

# Table of Contents

<b>1. INTRODUCTION .....</b>	<b>7</b>
1.1. Related Documentation .....	7
1.2. Definitions.....	7
1.2.1. Glossary .....	7
<b>2. REQUIREMENTS XML EDITION IMPLEMENTATION OVERVIEW.....</b>	<b>8</b>
2.1. CSE Database Extraction.....	8
<b>3. REQUIREMENTS XML EDITION GENERAL BEHAVIOR.....</b>	<b>9</b>
3.1. XML File build.....	9
3.1.1. XML Schema .....	9
3.1.2. XML Schema – Added attributes .....	13
3.1.3. XML Schema – Specific update .....	13
3.1.4. XML file .....	13
3.1.5. XML file edition specification .....	13
3.2. XML File edition.....	13
3.2.1. Requirements XML Edition with form.....	13
3.2.2. Form design .....	13
3.2.3. Expand collapse behavior .....	13
3.2.4. Node Modification.....	14
3.2.5. Structure Modification .....	14
3.3. XML File import.....	14
3.3.1. XML File save as .....	14
3.3.2. XML File save .....	14
3.4. CSE Database Update .....	14
3.4.1. Profile card Update .....	14
3.4.2. Unlock InfoPath form .....	14
3.4.3. Sub tree creation .....	14
3.4.4. Hierarchical Link deletion .....	14
3.4.5. Reordering .....	14
3.4.6. Update and performance .....	14
3.4.7. GUI refresh .....	15
3.4.8. Changing Rights .....	15
<b>4. APPENDIX A - IMPLEMENTATION SETTINGS.....</b>	<b>16</b>
4.1. Administration settings dependencies.....	16
4.1.1. XML File build .....	16
4.1.1.1. Without admin settings .....	16
4.1.1.2. Needed Attributes .....	16
4.1.1.3. Xhtml Attributes .....	17
4.1.1.4. Link attributes .....	19

4.1.1.5.	System Attributes.....	20
4.1.2.	XML File edition .....	20
4.1.2.1.	InfoPath Form association .....	20
4.1.2.2.	File name composition .....	20

## 5. APPENDIX B - TROUBLESHOOTING .....21

5.1.	Debug mode.....	21
5.1.1.	Declare .....	21
5.1.2.	File storage.....	21
5.1.3.	File usage .....	21

# List of figures and tables

---

Table 1 - Acronyms.....	7
-------------------------	---

# 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 RX9.

This document is divided into the following sections:

- Requirements XML Edition implementation overview
- Requirements XML Edition General Behavior
- Appendix A - Implementation settings
- Appendix B - Troubleshooting

## 1.1. 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

## 1.2. Definitions

---

### 1.2.1. Glossary

Acronym	Definition
BPA	Business Process Accelerator
CSE	Collaborative System Engineering

Table 1 - Acronyms

## 2. Requirements XML Edition implementation overview

---

### 2.1. *CSE Database Extraction*

---

CSE\_XML module is started by user interface selecting a Root Node and clicking edit module or chapter short cut button. That feature is available for requirements and IVVQ trees. The entire hierarchical sub tree stored in the database is retrieved according to the sorting displayed in the CSE GUI.

Then some operation of filtering can be conditioned by admin setting as is described hereafter. The default behavior retrieves the entire sub tree.



## 3. Requirements XML Edition General Behavior

---

### 3.1. XML File build

---

#### 3.1.1. XML Schema

Extracted information is stored in an XML format according to a generic XML schema. This schema expects nodes named by the class names used in the CSE data model.

In our example, the schema has ***“Requirements module”*** as root and can have sub elements

- ***Primary\_Requirement***
- ***Derived\_Requirement***
- ***Requirements\_Information***
- ***Requirements\_Chapter***

The entire schema has the same structure as in the CSE data model

Each node of the schema contains some attributes declared in each class. Some are mandatory and others are optional. That is customized thru InfoPath schema. Below is presented the CSE standard schema.

◆ E	Requirements_Module	(Requirements_Module)
G	(group1)	choice
A	ref=ns1:Form_Object_Attributes	
A	ref=ns1:ST_Object_Attributes	
A	ref=ns1:CSE_Container_Object_Attribu	

◆ G	(group1)	choice
E	ref=ns1:Primary_Requirement	
E	ref=ns1:Derived_Requirement	
E	ref=ns1:Requirements_Information	
E	ref=ns1:Requirements_Chapter	

◆ A	ref=ns1:Form_Object_Attribute
-----	-------------------------------

◆ A	ref=ns1:ST_Object_Attributes
-----	------------------------------

◆ A	ref=ns1:CSE_Container_Object
-----	------------------------------






◆ E	Requirements_Chapter	(Requirements_Chapter)
G	(group1)	choice
A	ref=ns1:Form_Object_Attributes	
A	ref=ns1:ST_Object_Attributes	
A	ref=ns1:CSE_Container_Object_Attribut	






◆ G	(group1)	choice
E	ref=ns1:Primary_Requirement	
E	ref=ns1:Derived_Requirement	
E	ref=ns1:Requirements_Information	
E	ref=ns1:Requirements_Chapter	(Requirements_Chapter)





◆ A	ref=ns1:Form_Object_Attribute
-----	-------------------------------

◆ A	ref=ns1:ST_Object_Attributes
-----	------------------------------

◆ A	ref=ns1:CSE_Container_Object
-----	------------------------------

◆ E	Primary_Requirement	(Primary_Requirement)
E	TDMX_REQUIREMENT	xhtmlType
	ref=ns1:Form_Object_Attributes	
	ref=ns1:ST_Object_Attributes	Primary_Requirement <xs:element>
	ref=ns1:CSE_Single_Object_Attributes	
	ref=ns1:CSE_Object_Requirement_Attributes	
	ref=ns1:CSE_Primary_Requirement_Attributes	

◆ E	Derived_Requirement	(Derived_Requirement)
E	TDMX_REQUIREMENT	xhtmlType
E	TDMX_SOURCE_REQUIREMENT	xhtmlType
	ref=ns1:Form_Object_Attributes	
	ref=ns1:ST_Object_Attributes	
	ref=ns1:CSE_Single_Object_Attributes	
	ref=ns1:CSE_Object_Requirement_Attributes	
	ref=ns1:CSE_Derived_Requirement_Attributes	

◆ E	Requirements_Information	(Requirements_Information)
E	TDMX_REQUIREMENT	xhtmlType
	ref=ns1:Form_Object_Attributes	
	ref=ns1:ST_Object_Attributes	
	ref=ns1:CSE_Single_Object_Attributes	
	ref=ns1:CSE_Requirement_Information_Attributes	

Form_Object_Attributes		
A	CHAPTERID	string
A	NEED_SYNC	string

Ct xhtmlType		
◆		
A	NEED_SYNC	string

ST_Object_Attributes		
A	OBJECT_ID	string
A	CLASS_ID	string
A	TDM_ID	string
A	REVISION	string
A	STATE	string
A	TDM_DESCRIPTION	string
A	GENERAL_LINK	string
A	USER_OBJECT_ID	string

CSE_Single_Object_Attributes		
A	COLLAPSE	boolean

CSE_Container_Object_Attributes		
A	COLLAPSE	boolean

CSE_Object_Requirement_Attributes		
A	TDMX_COMMENTS	string
A	TDMX_DESIGN_CONSTRAINT	string
A	TDMX_EXTERNAL_ID	string
A	TDMX_OWNER	string
A	TDMX_PRIORITY	string
A	TDMX_MATURITY_STAGE	string
A	TDMX_RISK_LEVEL	string
A	TDMX_SCOPE	string
A	TDMX_TYPE	string
A	TDMX_COMPLIANCE_LEVEL	string
A	TDMX_CLASS	string
A	TDMX_SOURCE_CATEGORY	string
A	TDMX_VERIFICATION_METHOD	string
A	TDMX_WEIGHT_FACTOR	string
A	TDMX_QUALITY	string
A	TDMX_VALUE_VALIDITY_COND	string

CSE_Requirement_Information_Attributes		
----------------------------------------	--	--

CSE_Primary_Requirement_Attributes		
------------------------------------	--	--

CSE_Derived_Requirement_Attributes		
------------------------------------	--	--

### **3.1.2. XML Schema – Added attributes**

Each node of the schema has got a few attributes needed for edition and database update after edition

- CHAPTERID
- COLLAPSE
- NEED\_SYNC

These are system attributes and automatically exported from Smarteam.

### **3.1.3. XML Schema – Specific update**

All the CSE data model attributes are not mapped to an XML node attribute. The schema contains some nodes that are not classes of the CSE data model.

For instance, the TDMX\_REQUIREMENT and TDMX\_SOURCE\_REQUIREMENT fields are mapped to a XML tree. Only one attribute is defined in the root tree. That is NEED\_SYNC.

### **3.1.4. XML file**

The XML file is built according to the ordered sub tree selected by user and retrieved from data base. Without any configuration declared in admin settings, all the attributes are filled in each node of the tree, and no attributes are treated as a node (like TDMX\_REQUIREMENT or TDMX\_SOURCE\_REQUIREMENT). Each node contains the CLASS\_ID and OBJECT\_ID attributes used for Database Update.

The systems attributes are filled with specific initial values and added to each node.

### **3.1.5. XML file edition specification**

Some specific information is added to the XML file to be edited in the selected XML editor. Each of them may expect different additional information. Today edition is only in Microsoft InfoPath program.

## **3.2. XML File edition**

---

### **3.2.1. Requirements XML Edition with form**

The edition of extracted information is done with InfoPath and a designed InfoPath Form called by a specific name. For Requirement tree this name is **ST:CSE\_REQTREE** and is defined by the script hook calling the edit module functionality. For Requirement chapter this name is **ST:CSE\_CHAPTER\_REQTREE** and is defined by the script hook calling the edit chapter functionality.

For IVVQ tree this name is **ST:CSE\_IVVQMODULE** and is defined by the script hook calling the edit module functionality. For IVVQ chapter this name is **ST:CSE\_IVVQCHAPTER** and is defined by the script hook calling the edit chapter functionality.

### **3.2.2. Form design**

The form used editing XML data is designed with client look and feel according to the XML schema described before. It embeds some code to modify the XML data of system attributes. This code is generic to each form and must be embedded during form design to update the NEED\_SYNC system attribute, store the CHAPTERID information, handle the TDMX\_LOCKED\_BY attribute for concurrent access and modify the COLLAPSE attribute value of the nodes.

### **3.2.3. Expand collapse behavior**

A checkbox is usable in each node that can embed a sub tree. The form must be designed to use this value.

### **3.2.4. Node Modification**

NEED\_SYNC attribute is used during import process so as to launch or not the synchronization process. The fields declared in the form as containing xhtml structure according the schema can store Rich-text, tables or pictures

### **3.2.5. Structure Modification**

Some new element can be added anywhere in the sub tree according to the contextual menu of each node.

## **3.3. XML File import**

---

### **3.3.1. XML File save as**

The user can save the form with a chosen name selecting the **<Save as menu item>**. This allows breaking working with CSE and recall the InfoPath saved file to continue edition.

### **3.3.2. XML File save**

The user can save the form in CSE by selecting the **<Save> menu item**. If Smarteam CSE is not alive, user is prompt to restart it and save again.

## **3.4. CSE Database Update**

---

### **3.4.1. Profile card Update**

For each element exported and modified in InfoPath, the database is updated.

### **3.4.2. Unlock InfoPath form**

In order to manage concurrent access to the same elements, a mechanism logs in the DB all the user ids that are editing any requirement object thru InfoPath. Those ids are removed automatically when the user closes InfoPath. Nevertheless a "Unlock" UDT is available to manually erase those ids from the DB.

### **3.4.3. Sub tree creation**

New elements created in InfoPath are added in the tree according to their position in the InfoPath form.

### **3.4.4. Hierarchical Link deletion**

Elements that are deleted in the InfoPath form are not deleted in the CSE database. Only The hierarchical links are deleted.

### **3.4.5. Reordering**

According to the modification in the InfoPath document the CSE tree is reordered

### **3.4.6. Update and performance**

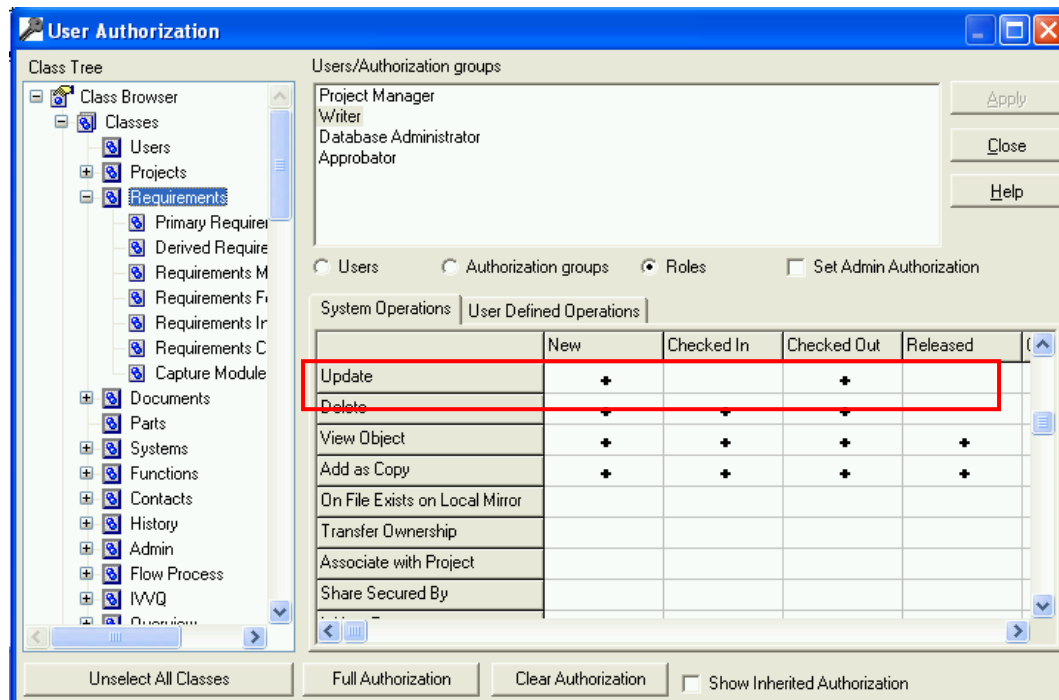
The Form stores information used in import phase about data modification and structure modification. The database is modified only in needed cases to optimize the performance in import

### 3.4.7. GUI refresh

After the import, the tree displayed in CSE is refreshed with the imported modifications.

### 3.4.8. Changing Rights

In agreement with the concurrent access functionality, it is forbidden to update any Checked In and Released objects. Consequently, you have to set the users rights as described here below. Those settings have to be applied for the IVVQ tree as well.



## 4. Appendix A - Implementation settings

### 4.1. Administration settings dependencies

#### 4.1.1. XML File build

##### 4.1.1.1. Without admin settings

Without any additional admin settings, all the attributes of classes retrieved by the drilldown extraction are stored in the node of the XML file. They are set as attributes of a node with the same name as they are defined in the class. The node name has got the name of class (all the blank characters are replaced by <\_> characters).

##### 4.1.1.2. Needed Attributes

User can declare for each class of the data model which attributes have to be retrieved. This declaration is done by a list of attributes in the long value of an admin setting like in the example below.

Add Admin Setting		
Section	NEEDED_ATTRIBUTES	Attributes to retrieve
Subject	PRIMARY_REQUIREMENT	<i>Class name of data model in uppercase and blank characters replaced by &lt;_&gt;</i>
Description	Declare attributes of the class that has to be retrieved	
Value1		
Value2		
Value3		
Value4		
LongField	OBJECT_ID;CLASS_ID;TDM_ID;TDM_DESCRIPTION;REVISION; TDMX_REQUIREMENT;STATE;TDMX_PRIORITY;TDMX_SCOPE; TDMX_SOURCE_CATEGORY;TDMX_DESIGN_CONSTRAINT; TDMX_EXTERNAL_ID;TDMX_COMMENTS;TDMX_OWNER; TDMX_MATURITY_STAGE;TDMX_TYPE;TDMX_COMPLIANCE_LEVEL; TDMX_CLASS;TDMX_VERIFICATION_METHOD; TDMX_WEIGHT_FACTOR;TDMX_QUALITY; TDMX_VALUE_VALIDITY_COND;TDMX_RISK_LEVEL; TDMX_LOCKED_BY	List of Attributes separated by ;
<Other>		

*The CLASS\_ID and the OBJECT\_ID must be present in the list of attributes*



Below are listed all the needed attributes per class:

- Requirements Chapter:  
OBJECT\_ID;CLASS\_ID;TDM\_ID;TDM\_DESCRIPTION;STATE;REVISION;TDMX\_LOCKED\_BY
- Requirements Information:  
OBJECT\_ID;CLASS\_ID;TDM\_ID;TDM\_DESCRIPTION;TDMX\_REQUIREMENT;REVISION;STATE;TDMX\_LOCKED\_BY
- Derived Requirement:  
OBJECT\_ID;CLASS\_ID;TDM\_ID;TDM\_DESCRIPTION;REVISION;TDMX\_REQUIREMENT;STATE;TDMX\_PRIORITY;TDMX\_SCOPE;TDMX\_SOURCE\_CATEGORY;TDMX\_DESIGN\_CONSTRAINT;TDMX\_EXTERNAL\_ID;TDMX\_COMMENTS;TDMX\_OWNER;TDMX\_MATURITY\_STAGE;TDMX\_TYPE;TDMX\_COMPLIANCE\_LEVEL;TDMX\_CLASS;TDMX\_VERIFICATION\_METHOD;TDMX\_WEIGHT\_FACTOR;TDMX\_QUALITY;TDMX\_VALUE\_VALIDITY\_CONDITION;TDMX\_RISK\_LEVEL;TDMX\_SOURCE\_REQUIREMENT;TDMX\_LOCKED\_BY
- Primary Requirement:  
OBJECT\_ID;CLASS\_ID;TDM\_ID;TDM\_DESCRIPTION;REVISION;TDMX\_REQUIREMENT;STATE;TDMX\_PRIORITY;TDMX\_SCOPE;TDMX\_SOURCE\_CATEGORY;TDMX\_DESIGN\_CONSTRAINT;TDMX\_EXTERNAL\_ID;TDMX\_COMMENTS;TDMX\_OWNER;TDMX\_MATURITY\_STAGE;TDMX\_TYPE;TDMX\_COMPLIANCE\_LEVEL;TDMX\_CLASS;TDMX\_VERIFICATION\_METHOD;TDMX\_WEIGHT\_FACTOR;TDMX\_QUALITY;TDMX\_VALUE\_VALIDITY\_CONDITION;TDMX\_RISK\_LEVEL;TDMX\_LOCKED\_BY
- Requirements Module:  
OBJECT\_ID;CLASS\_ID;TDM\_ID;TDM\_DESCRIPTION;REVISION;STATE;TDMX\_LOCKED\_BY
- IVVQ Module:  
OBJECT\_ID;CLASS\_ID;CN\_ID;TDMX\_NAME;REVISION;STATE;TDMX\_IVVQ\_COMMENT;TDMX\_LOCKED\_BY
- IVVQ Chapter:  
OBJECT\_ID;CLASS\_ID;CN\_ID;TDMX\_NAME;REVISION;STATE;TDMX\_IVVQ\_COMMENT;TDMX\_LOCKED\_BY
- IVVQ Sheet:  
OBJECT\_ID;CLASS\_ID;CN\_ID;TDMX\_NAME;REVISION;STATE;TDMX\_MATURITY\_STAGE;TDMX\_MOV;TDMX\_TEST\_INPUT;TDMX\_VERIFICATION\_STAGE;TDMX\_MOV\_DESCRIPTION;TDMX\_TEST\_PROC\_DESC;TDMX\_LOCKED\_BY
- IVVQ Result:  
OBJECT\_ID;CLASS\_ID;CN\_ID;TDMX\_NAME;REVISION;STATE;TDMX\_RUN\_DATE;TDMX\_TESTER\_NAME;TDMX\_GLOBAL\_TEST\_RES;TDMX\_TR\_PATHNAME;TDMX\_TR\_FILENAME;TDMX\_IVVQ\_COMMENT;TDMX\_MATURITY\_STAGE;TDMX\_TEST\_BED;TDMX\_TEST\_OBJECTIVES;TDMX\_TEST\_RESULT\_DESC;TDMX\_LOCKED\_BY
- IVVQ Matrix Item:  
OBJECT\_ID;CLASS\_ID;CN\_ID;TDMX\_NAME;REVISION;STATE;TDMX\_MATURITY\_STAGE;TDMX\_VERIFICATION\_STATUS;TDMX\_LOCKED\_BY
- IVVQ Information:  
OBJECT\_ID;CLASS\_ID;CN\_ID;TDMX\_NAME;REVISION;STATE;TDMX\_INFORMATION;TDMX\_LOCKED\_BY
- IVVQ Analysis:  
OBJECT\_ID;CLASS\_ID;CN\_ID;TDMX\_NAME;REVISION;STATE;TDMX\_MATURITY\_STAGE;TDMX\_CONTENT\_DATA;TDMX\_LOCKED\_BY

### 4.1.1.3. Xhtml Attributes

User can declare for each class of the data model which attributes have to be treated with XHTML content. This declaration is done by a list of attributes in the long value of an admin setting like in the example above.

Add Admin Setting		
Section	XHTML_ATTRIBUTES	XHTML Attributes
Subject	PRIMARY_REQUIREMENT	<b><i>Class name of data model in uppercase and blank characters replaced by &lt;_&gt;</i></b>
Description	Declare attributes of the class that has to be treated with xhtml content	
Value1		
Value2		
Value3		
Value4		
LongField	TDMX_REQUIREMENT	List of Attributes separated by ;
<Other>		

*Those Attributes will be not integrated in the XML file as attributes of a node but as a sub element which name will be the attributes name. They will contain sub elements with xhtml meaning.*

*Those nodes, like <TDMX\_REQUIREMENT>, will not have attributes except system attributes.*

Add Admin Setting		
Section	XHTML_ATTRIBUTES	XHTML Attributes
Subject	IVVQ_RESULT	<b><i>Class name of data model in uppercase and blank characters replaced by &lt;_&gt;</i></b>
Description	Attributes for a IVVQ Result to treat as XHTML	
Value1		
Value2		
Value3		
Value4		
LongField	TDMX_TEST_OBJECTIVES;TDMX_TEST_RESULT_DESC	List of Attributes separated by ;
<Other>		

Add Admin Setting		
Section	XHTML_ATTRIBUTES	XHTML Attributes
Subject	IVVQ_SHEET	<b><i>Class name of data model in uppercase and blank characters replaced by &lt;_&gt;</i></b>
Description	Attributes for a IVVQ Sheet to treat as XHTML	
Value1		
Value2		
Value3		
Value4		
LongField	TDMX_MOV_DESCRIPTION;TDMX_TEST_PROC_DESC; TDMX_TEST_INPUT	List of Attributes separated by ;
<Other>		

Add Admin Setting		
Section	XHTML_ATTRIBUTES	XHTML Attributes
Subject	IVVQ_INFORMATION	<i>Class name of data model in uppercase and blank characters replaced by &lt;_&gt;</i>
Description	Attributes for a IVVQ Information to treat as XHTML	
Value1		
Value2		
Value3		
Value4		
LongField	TDMX_INFORMATION	List of Attributes separated by ;
<Other>		

Add Admin Setting		
Section	XHTML_ATTRIBUTES	XHTML Attributes
Subject	IVVQ_ANALYSIS	<i>Class name of data model in uppercase and blank characters replaced by &lt;_&gt;</i>
Description	Attributes for a IVVQ Analysis to treat as XHTML	
Value1		
Value2		
Value3		
Value4		
LongField	TDMX_INFORMATION	List of Attributes separated by ;
<Other>		

#### 4.1.1.4. Link attributes

So as to be able to display the objects linked to a matrix item, you need to define two more admin settings as below:

Add Admin Setting	
Section	DRILL_DOWN_LINK_ALLOW_CLASSES
Subject	IVVQ_REQUIREMENTS
Description	Declare both classes of the relationship
Value1	IVVQ
Value2	REQUIREMENT
Value3	
Value4	

That admin setting allows displaying into InfoPath the attributes of the relationship IVVQ-Requirements.

Add Admin Setting	
Section	DRILL_DOWN_LINK_ALLOW_CLASSES
Subject	IVVQ_IVVQ
Description	Declare both classes of the relationship
Value1	IVVQ
Value2	IVVQ
Value3	
Value4	

That admin setting allows displaying into InfoPath the attributes of the relationship IVVQ-IVVQ.

#### 4.1.1.5. System Attributes

A few system attributes need to be added in each nodes of the XML file produced.

**NEED\_SYNC**  
**CHAPTERID**  
**OBJECT\_ID**  
**CLASS\_ID**  
**TDM\_ID**  
**REVISION**  
**STATE**  
**TDM\_DESCRIPTION**  
**COLLAPSE**

For primary and derived requirements the following attributes are mandatory too:

**TDMX\_MATURITY\_STAGE**  
**TDMX\_TYPE**  
**TDMX\_COMPLIANCE\_LEVEL**  
**TDMX\_SOURCE\_CATEGORY**

For **Xhtml Attributes** which are stored as sub node, just **NEED\_SYNC** attribute is added to the node.

### 4.1.2. XML File edition

#### 4.1.2.1. InfoPath Form association

The XML file that contains the exported values of CSE database is merged with information needed by InfoPath to open it with a declared Form. In the case of Requirements Module edition the form name is **ST:CSE\_REQTREE** , for Requirements Chapter edition the form name is **ST:CSE\_CHAPTER\_REQTREE**, for IVVQ module the form name is **ST;CSE\_IVVQMODULE** and for IVVQ Chapter, the form name is **ST:CSE\_IVVQCHAPTER**.

There is no admin setting to declare it. This name is hard coded in the script file called from the script maintenance hook declared for this functionality.

#### 4.1.2.2. File name composition

The file name of the XML data is a composition of

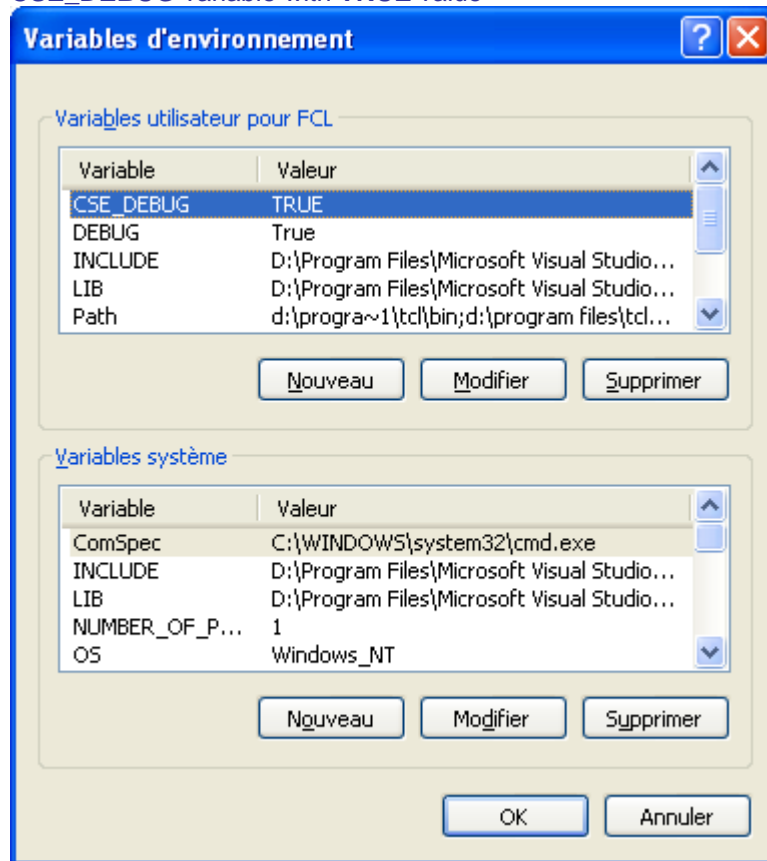
**SMARTTEAM-CSE-<TDM\_ID of root node selected by user>\_<Revision of root node selected by user>.xml**

## 5. Appendix B - Troubleshooting

### 5.1. Debug mode

#### 5.1.1. Declare

If you want to store information usable for debugging you need to declare an environment **CSE\_DEBUG** variable with **TRUE** value



#### 5.1.2. File storage

The file that store those debug information is names **cse.log**.  
Is stored in **C:\Documents and Settings\<user>\Local Settings\Temp**

#### 5.1.3. File usage

This file is used in each stage of the export and the import. Each line is composed by a tag and a value separated by a tag. This value is a time stamp of execution.

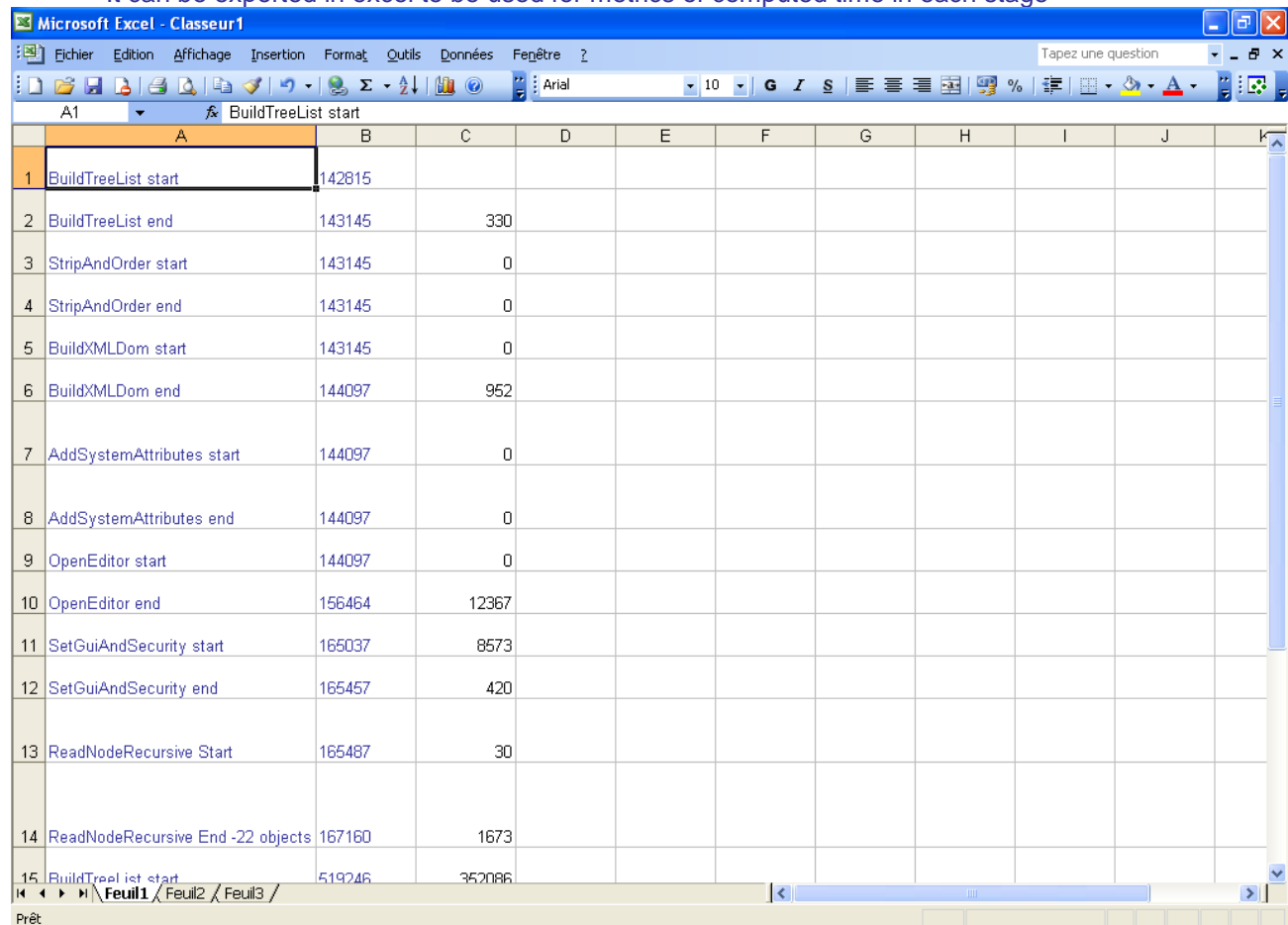
```
----
BuildTreeList start      142815
BuildTreeList end        143145
StripAndOrder start      143145
StripAndOrder end        143145
BuildXMLDom start        143145
BuildXMLDom end          144097
AddSystemAttributes start 144097
```

```

AddSystemAttributes end      144097
OpenEditor start            144097
OpenEditor end 156464
SetGuiAndSecurity start 165037
SetGuiAndSecurity end 165457
ReadNodeRecursive Start      165487
ReadNodeRecursive End -22 objects 167160
BuildTreeList start          519246
BuildTreeList end            519456
StripAndOrder start          519456
StripAndOrder end            519456
BuildXMLDom start            519456
BuildXMLDom end              520077

```

It can be exported in excel to be used for metrics of computed time in each stage



	A	B	C	D	E	F	G	H	I	J	K
1	BuildTreeList start	142815									
2	BuildTreeList end	143145	330								
3	StripAndOrder start	143145	0								
4	StripAndOrder end	143145	0								
5	BuildXMLDom start	143145	0								
6	BuildXMLDom end	144097	952								
7	AddSystemAttributes start	144097	0								
8	AddSystemAttributes end	144097	0								
9	OpenEditor start	144097	0								
10	OpenEditor end	156464	12367								
11	SetGuiAndSecurity start	165037	8573								
12	SetGuiAndSecurity end	165457	420								
13	ReadNodeRecursive Start	165487	30								
14	ReadNodeRecursive End -22 objects	167160	1673								
15	BuildTreeList start	519246	352086								