



SmartDX

ENOVIA SmarTeam® integrated data export

Implementation Guide

BPA Delivery 8 for V5R20 (V5.8)

Version: V5R20 (V5.8)

Date: 30.11.2007

Modified: 28.01.2010

Table of content

| | | |
|----------|--|-----------|
| 1 | Introduction | 6 |
| 1.1.1 | Related Documentation | 6 |
| 1.1.2 | Before Implementing | 6 |
| 2 | Database customization | 7 |
| 2.1 | Upgrading the WizSrc | 7 |
| 2.2 | Data Model Designer | 8 |
| 2.2.1 | Assign class mechanisms | 9 |
| 2.2.2 | Set composition for Customer and contact person | 14 |
| 2.2.3 | Enable 'Folder structure' capability..... | 16 |
| 2.3 | Form Designer | 19 |
| 2.4 | Script maintenance | 24 |
| 2.5 | Menu editor | 27 |
| 3 | Administration | 29 |
| 3.1 | 3.1 Admin Tool..... | 29 |
| 3.1.1 | Purpose | 29 |
| 3.1.2 | Access privileges for the Admin Tool | 29 |
| 3.1.3 | Set up the Admin Tool | 29 |
| 3.1.4 | The Admin Tools initialization parameters | 33 |
| 3.2 | Working with the Admin Tool | 40 |
| 3.2.1 | Add chapter | 40 |
| 3.2.2 | Add a new parameter | 40 |
| 3.2.3 | Change the value of a parameter..... | 42 |
| 3.2.4 | Save changes in the Admin Tool..... | 43 |
| 3.3 | Initialization file (Server side) | 43 |
| 3.3.1 | SmartDX_CallFunctions.ini..... | 43 |
| 3.4 | Language files..... | 46 |
| 3.5 | Understanding the exchange root directory..... | 46 |
| 3.6 | Defining the SmartDX server side behavior..... | 48 |
| 3.7 | Set up the CATIA environment for SmartDX server side batch process | 53 |
| 3.8 | Set up CATIA Standard Attributes | 54 |
| 3.9 | Set up CATIA additional attributes..... | 56 |
| 3.10 | Set up CATIA additional parameter | 57 |
| 3.11 | Set up CATIA renaming rules | 57 |
| 3.11.1 | Renaming through concatenating SmarTeam attributes..... | 57 |

| | | |
|----------|---|-----------|
| 3.11.2 | Individual defined renaming rules..... | 59 |
| 3.12 | Setup Mail transfer..... | 59 |
| 3.13 | XML Structure File | 60 |
| 3.13.1 | XML naming rules for additional SmarTeam attributes | 63 |
| 3.14 | Setup the daemon..... | 63 |
| 3.15 | Customizing capabilities with programmable ‚Plug Ins‘ | 64 |
| 3.16 | Customizing the different transfer modes | 66 |
| 3.16.1 | Hide or show different transfer modes to the user | 66 |
| 3.16.2 | Define Error and abnormal termination behavior | 67 |
| 4 | SmartDX enhancements | 68 |
| 4.1 | Customizing the SmartDX Export structure functionality | 68 |
| 4.2 | Exclude customer / contact person from SmartDX wizard list..... | 71 |
| 4.3 | Customer specific form in SmartDX wizard | 72 |
| 4.4 | Disable link between data ex. Obj. and transmittal or custom obj. | 74 |
| 5 | What’s new | 75 |
| 5.1 | DROP 6: CATIA Product to Part (All CAT Part) generation | 75 |
| 5.2 | DROP 6: Exporting CATIA documents related to DL names | 76 |
| 5.3 | DROP 6: Multi attribute mapping for CATIA properties / parameter..... | 78 |
| 5.4 | DROP 7: Observe the server site process on client site | 78 |
| 5.5 | DROP 7: New ‘Plug In’ to react on error interruptions..... | 79 |
| 5.6 | DROP 7: ‘ Plug In’ BeforeCATIABatch has been enhanced to build custom specific CATIA start strings | 80 |
| 5.7 | DROP 7: CATIA batch process interruption after predefined idle time..... | 80 |
| 5.8 | DROP 7: Enable users to copy all selected files in one step..... | 80 |
| 5.9 | DROP 8: Bugfixing and enhancements | 82 |
| 5.9.1 | SmartDX Monitor enhancement..... | 82 |
| 5.9.2 | Using same string multiple times for renaming CATIA documents | 82 |
| 5.9.3 | Disable the use of ‘copy all’ Check box after using the ‘Edit’ – Button ... | 82 |
| 6 | SmartDX flowchart | 83 |
| 7 | Known Issues..... | 85 |
| 7.1 | Defining formats for neutral conversion | 85 |
| 7.2 | Using ‘AllCatPart’ conversion..... | 85 |
| 7.3 | Using ‘AllCatPart’ in R19..... | 85 |
| 7.4 | CATIA batch process interrupts when CATIA Parts including Sheet Metal Design features | 85 |

Update report

| Chapter | Date | Editor | Modification |
|---------|----------|--------------|--------------------------------------|
| All | 30.11.07 | Michael Hitz | Origin version |
| 2.1 | 08.04.08 | Michael Hitz | Bug fixing for Drop 5 Service Pack 4 |
| All | 31.07.08 | Michael Hitz | Revised for Drop 6 |
| All | 19.05.09 | Michael Hitz | Revised for Drop 7 |
| 5.9 | 28.01.10 | Michael Hitz | Revised for Drop 8 |

Copyright notice

© 2010 DASSAULT SYSTEMES, 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 SYSTEMES.

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 SYSTEMES.

It shall be returned to DASSAULT SYSTEMES upon request.

DASSAULT SYSTEMES is a registered trademark of DASSAULT SYSTEMES.

All other trademarks belong to their respective owners.

ENOVIA SmarTeam® is a registered trademark of ENOVIA SmarTeam® Corporation Ltd.

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 / customization procedure for the SmartDX Business Process Accelerator (BPA).

It contains both, the client site implementation as well as the server site implementation.

1.1.1 Related Documentation

- SmartDX – Installation Guide
- SmartDX – User Guide

- 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

1.1.2 Before Implementing

Notes

Please refer to the installation guide before implement / customize SmartDX

When customizing, make sure that you have administrator privileges.

Close all other applications on the computer before proceeding with the installation.

2 Database customization

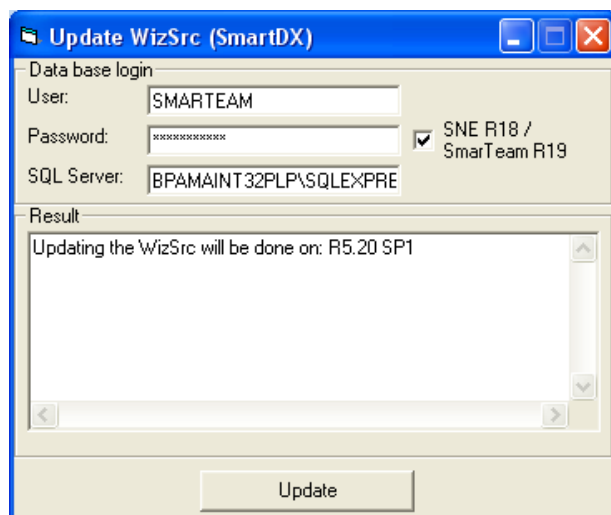
2.1 Upgrading the WizSrc

The SmartDX application is based on so called 'database level behavior' and 'Class level behavior'. These behaviors provide mechanisms used in the SmartDX application. To enable the use of SmartDX the Administrator has to check the appropriate mechanisms (see the next chapter for more details) during the data model designer. In order to provide these mechanisms the so called WizSrc database has to be upgraded. Therefore the installation provides an upgrade tool to perform the upgrade.

You can run the 'UdateWizSrcSmartDX.exe' from within the installation folder 'Implementation Tools'.

**It is strongly recommended to back up the origin WizSrc before preceding the upgrade tool.
For every upgrade of the WizSrc use the origin WizSrc. DO NOT UPGRADE A WIZSRC ALREADY UPGRADED BY SMARTDX PREVIOUSLY!**

REMARK: There is no need to upgrade the database again in case you did it with SmartDX Drop 5 or Drop 6



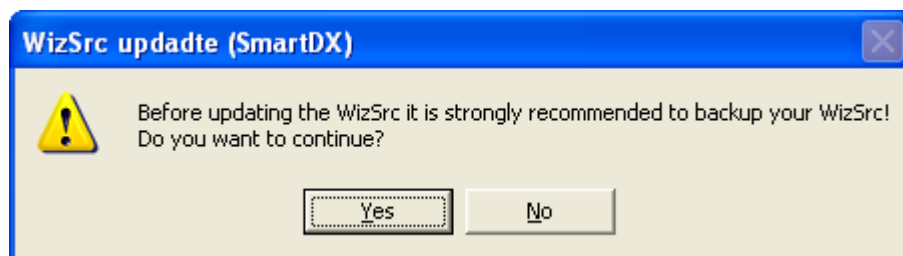
After starting the tool the window shows the current release and service pack of ENOVIA SmarTeam® you are using.

Dependent on the SmarTeam release you are using you are prompted to type in the database instance name and the password for the ENOVIA SmarTeam® database user.

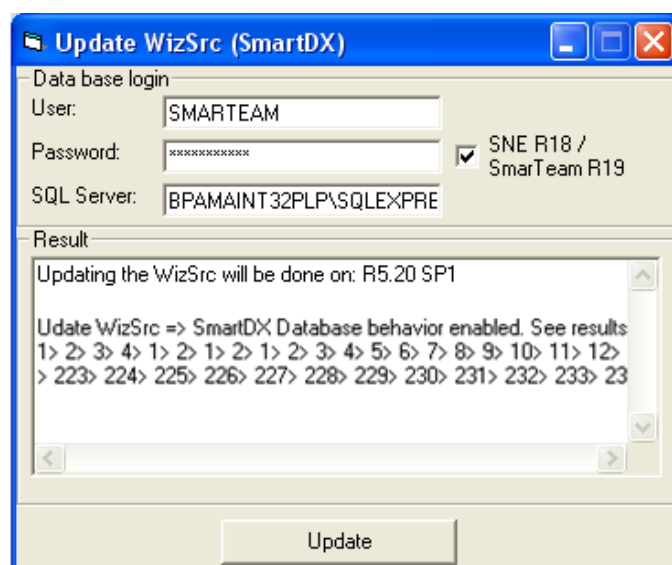
REMARK: In case your database is based on R19 or above please ensure you checked the option SNE R18 / SmarTeam R19.

After you typed in the instance name and the password perform the update button in order to update the WizSrc.

A warning message is asking for a back up of the origin WizSrc database. In case the origin database has been backed up you can precede using the 'Yes' button.



During runtime the result window is filled up with statements coming from the upgrade process. After the upgrade process finished refer to the result window to ensure the process was successful. In case the upgrade process finished successfully the result window only displays a set of numbers separated by an arrow sign '>'.

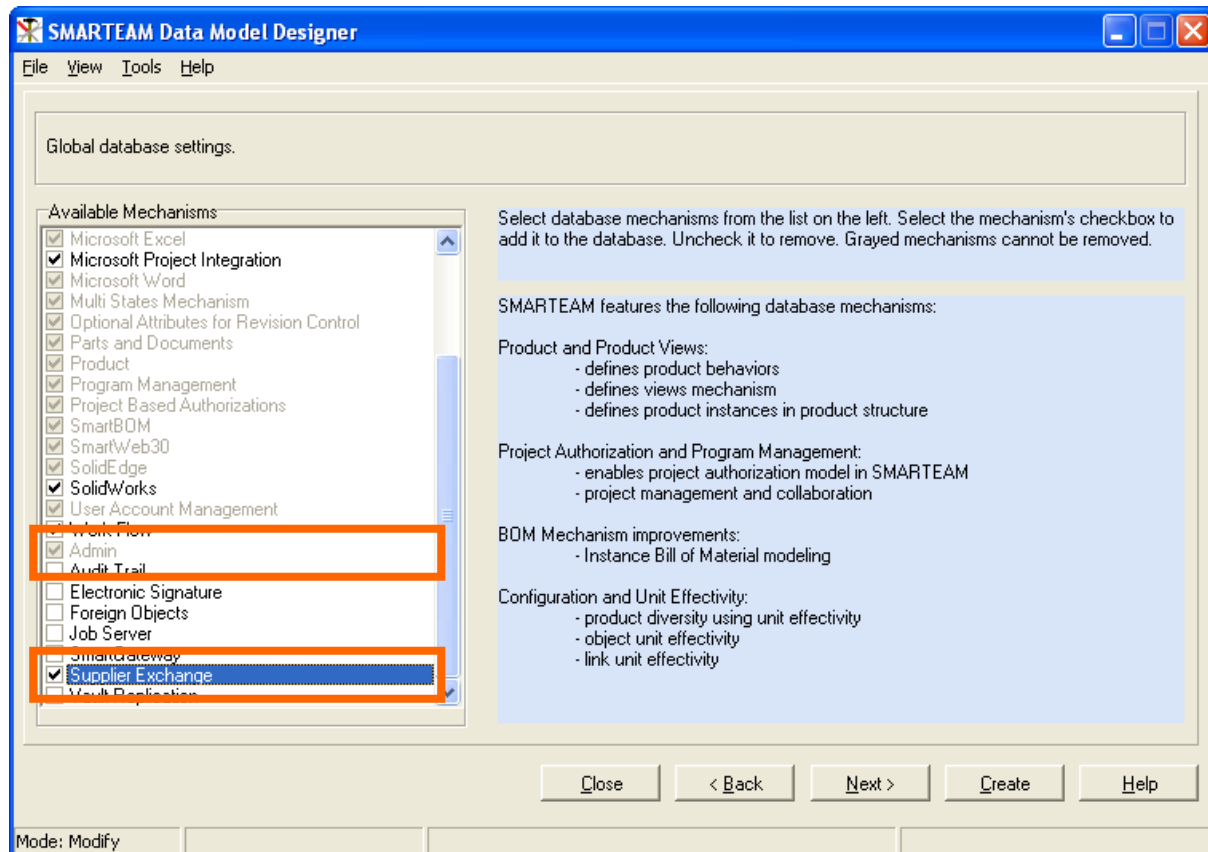


2.2 Data Model Designer

In order to enable the usage of SmartDX on the target ENOVIA SmarTeam® database you have to run the Data Model Designer (DMD). During the DMD process the SmartDX mechanisms have to be checked on the appropriate classes. This creates the needed Attributes and link classes automatically.

Please refer to the *ENOVIA SmarTeam® - Editor Administrator Guide* for a detailed description of the *Data Model Designer*.

It is strongly recommended to back up the origin database before preceding the Data model designer.



After launching the DMD select the 'Modify Database structure' in the File menu. In the next step you check the Mechanism 'Supplier Exchange' from the 'Available Mechanism' panel. Checking this mechanism automatically checks the 'Admin' mechanism, which is necessary to run SmartDX proper.

It might be that the 'Admin' behaviour on database level behaviour is not checked automatically. Please make sure the 'Admin' behaviour is checked.

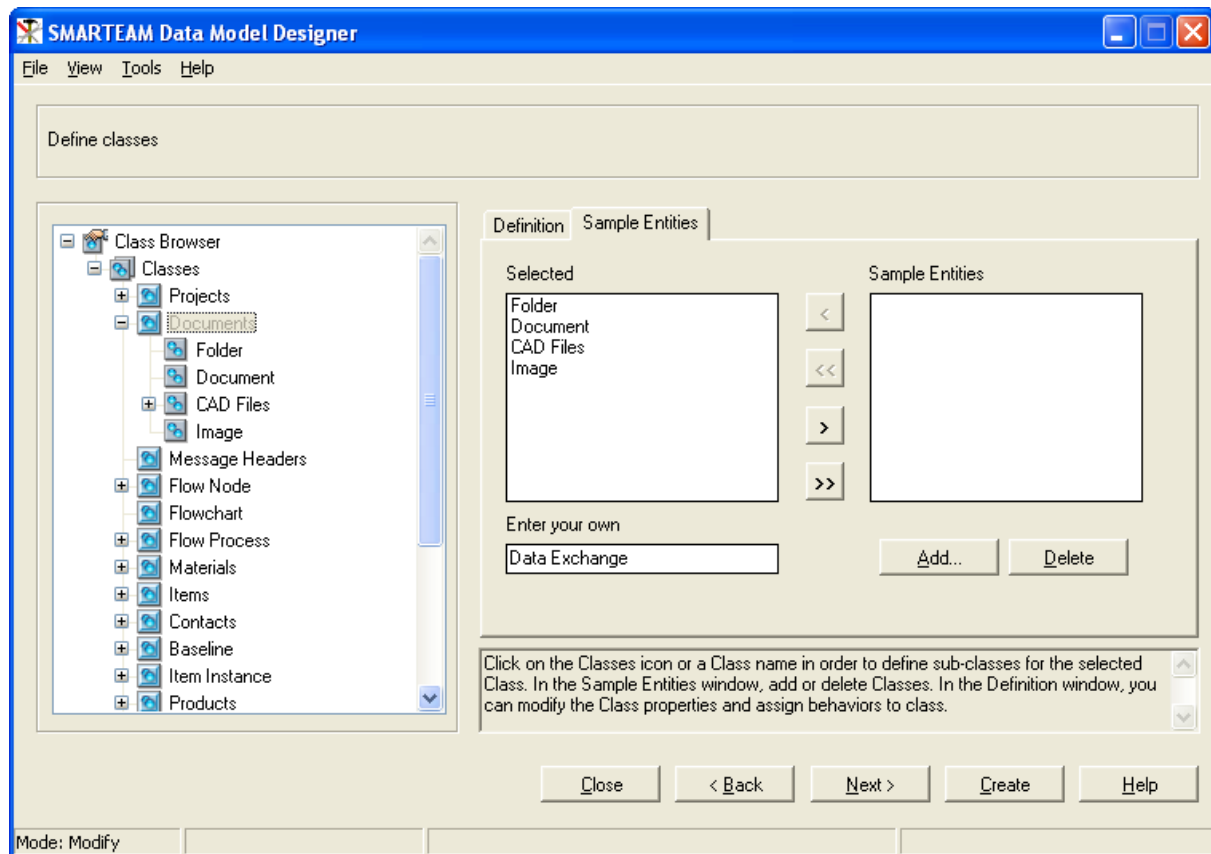
2.2.1 Assign class mechanisms

The application SmartDX is based on ENOVIA SmarTeam® class level behaviours. Therefore the following class mechanisms have to be assigned to the appropriate classes.

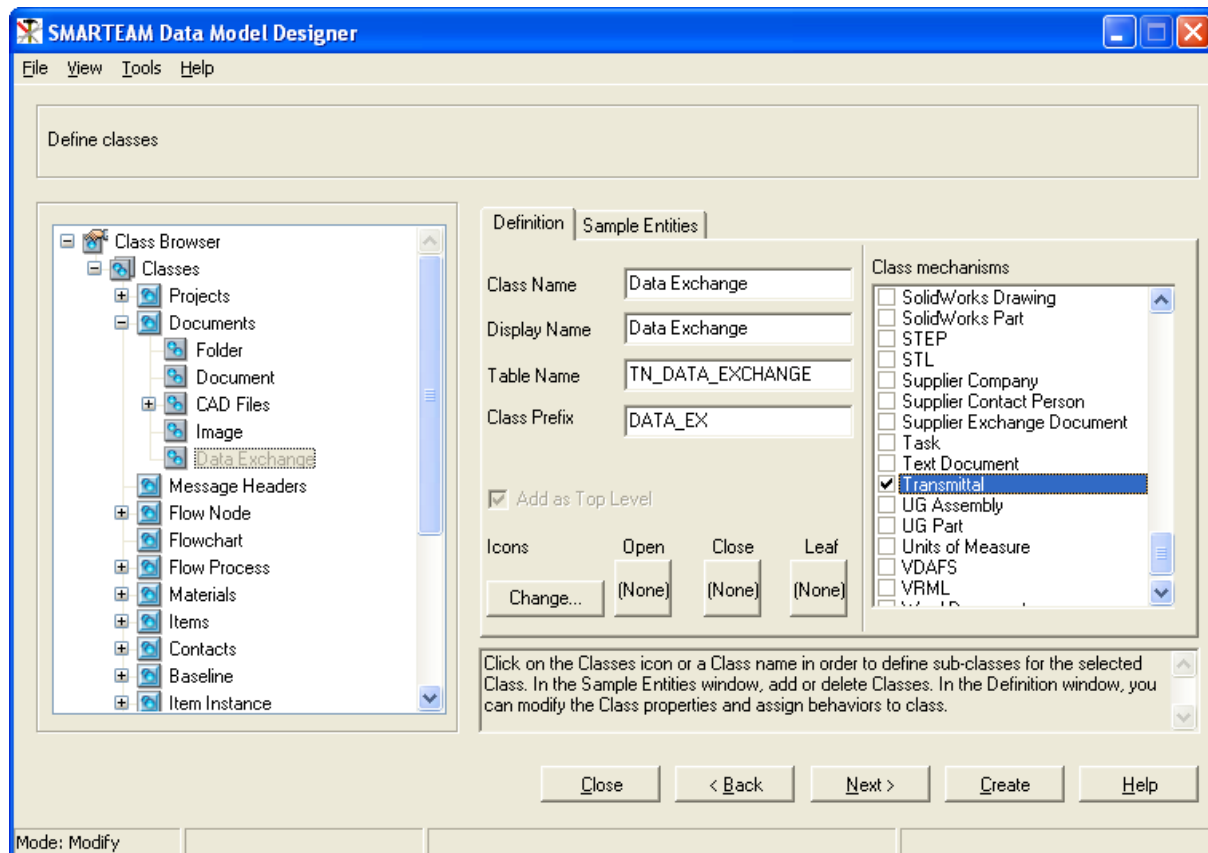
2.2.1.1 Transmittals

The transmittals are a kind of object that represents a data exchange process. It contains among others the information: who sent what at which time to whom.

It is advised to create a new class for this kind of objects.



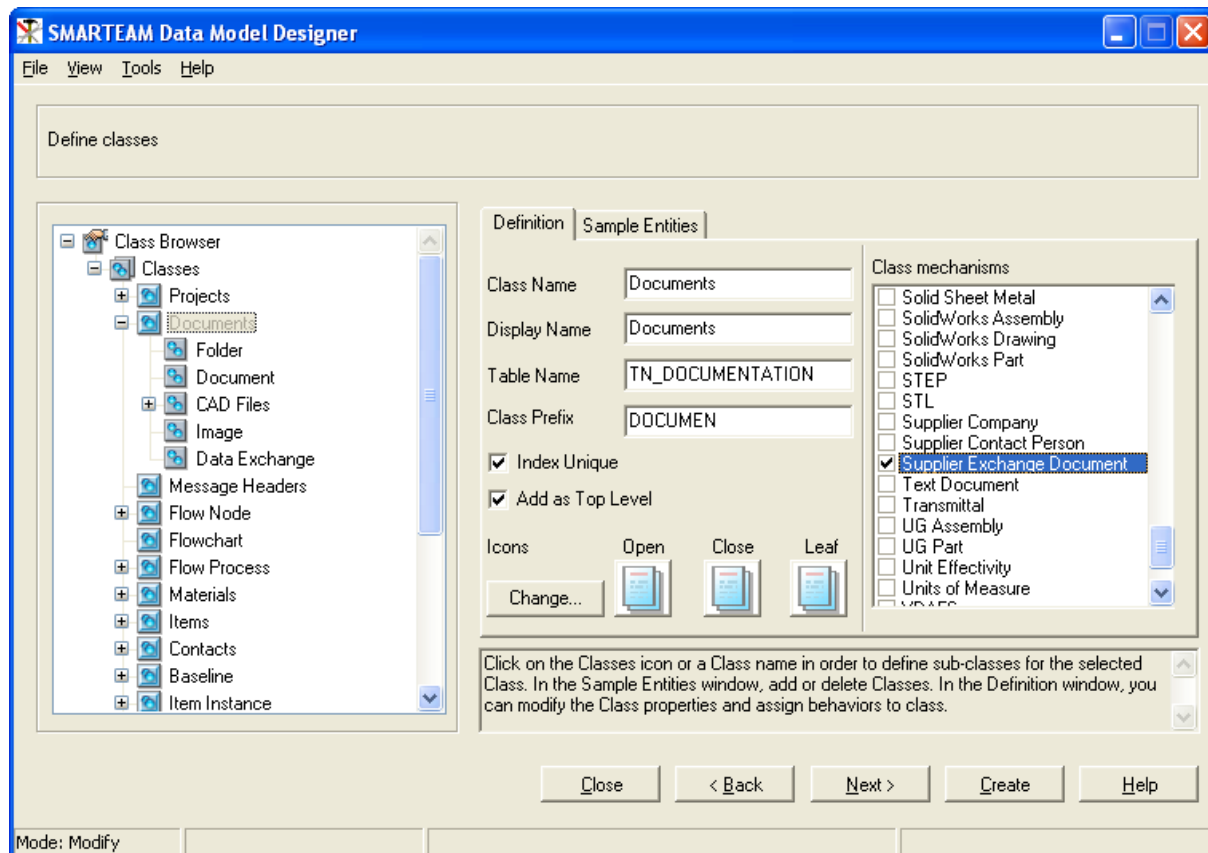
In the next step assign the class mechanism to it



All required attributes and link classes will be created automatically.

2.2.1.2 Supplier exchange documents

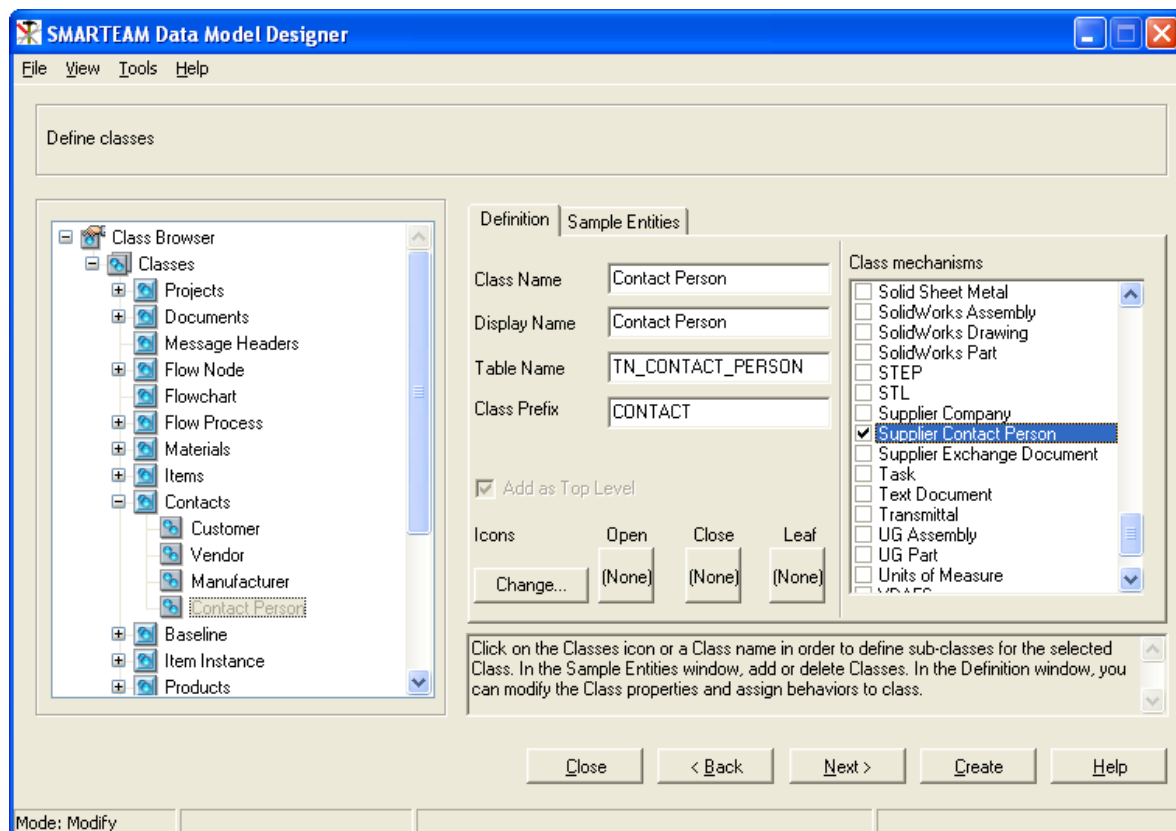
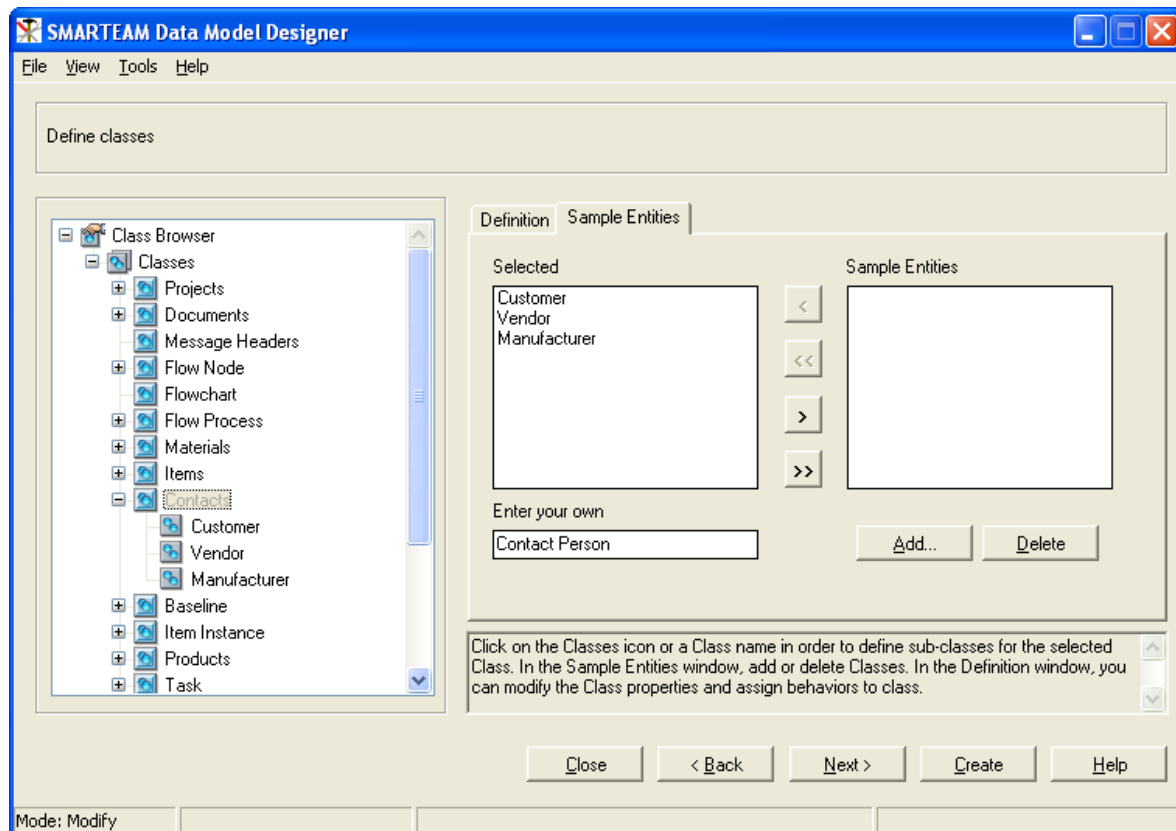
The 'supplier exchange documents' represents all kind of objects that can be transferred to a partner (i. e. CATIA Product, Word document). Therefore it is recommended to check the super class 'Documents' to ensure all kind of documents can be used for transfer.

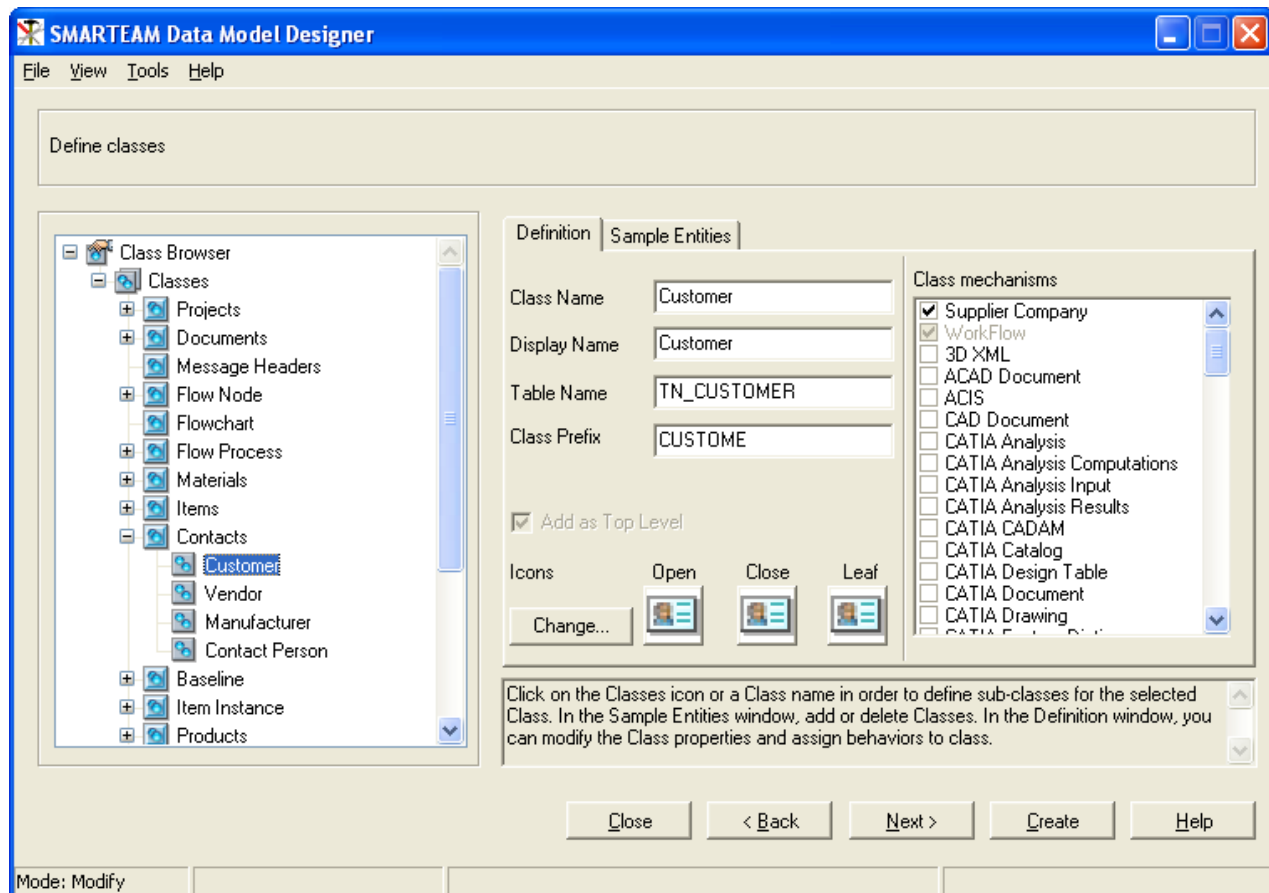


All required attributes and link classes will be created automatically.

2.2.1.3 Recipient (Customer / Contact person)

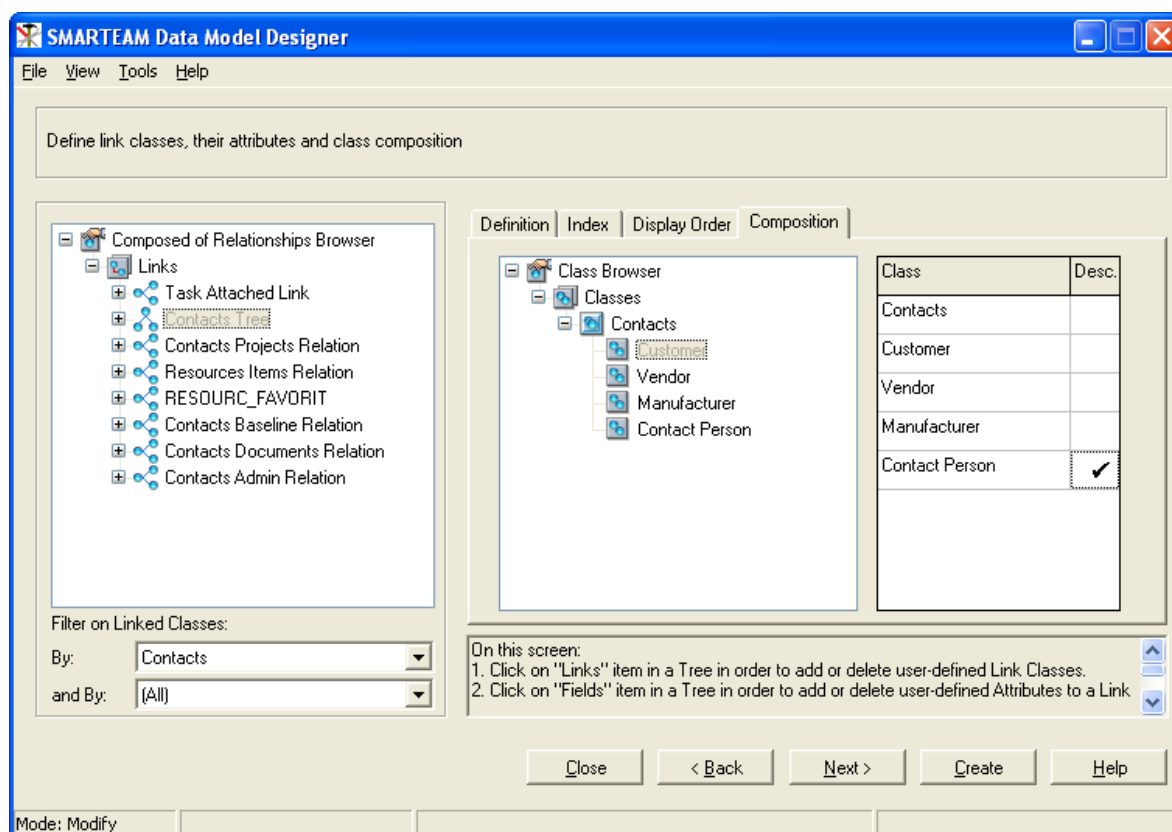
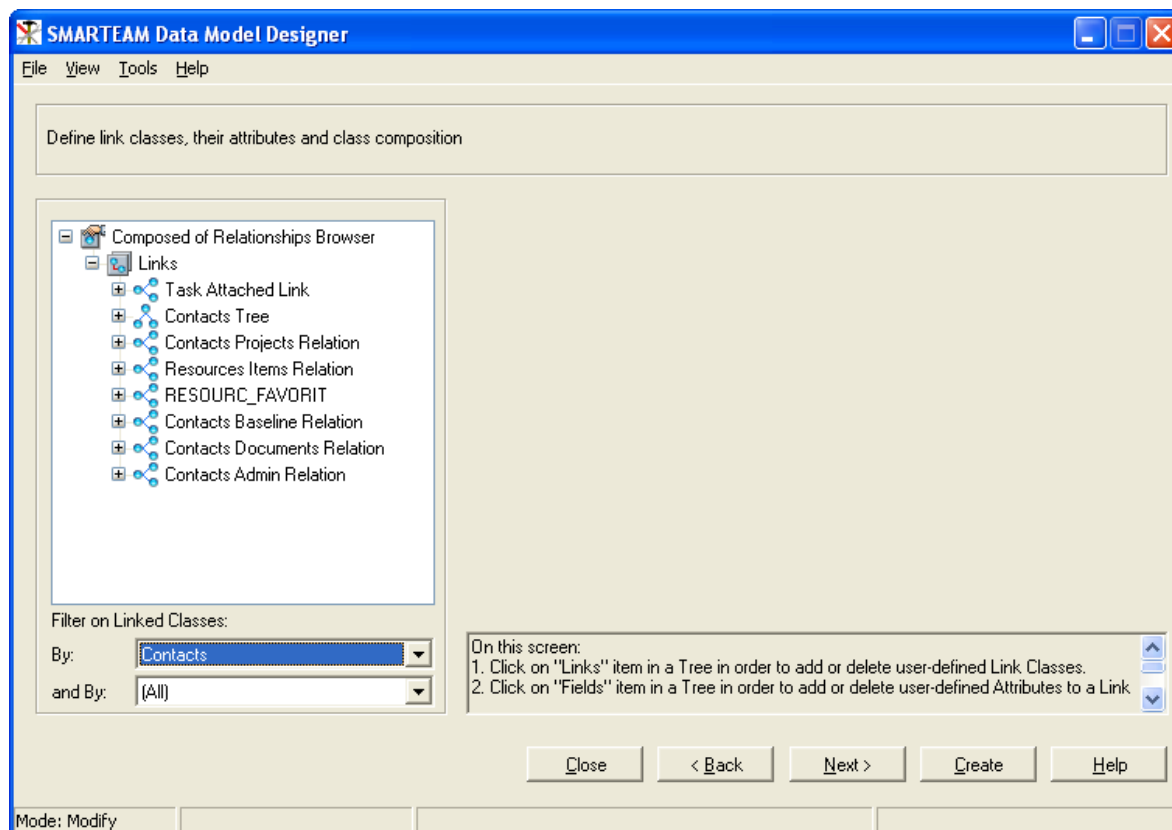
During the SmartDX wizard on the client side the user will be prompted to select a recipient for the transmittal process. Therefore you have to identify a customer, respectively a company class and a contact person class. These two classes contain predefinitions about transfer mode and conversion requirements.





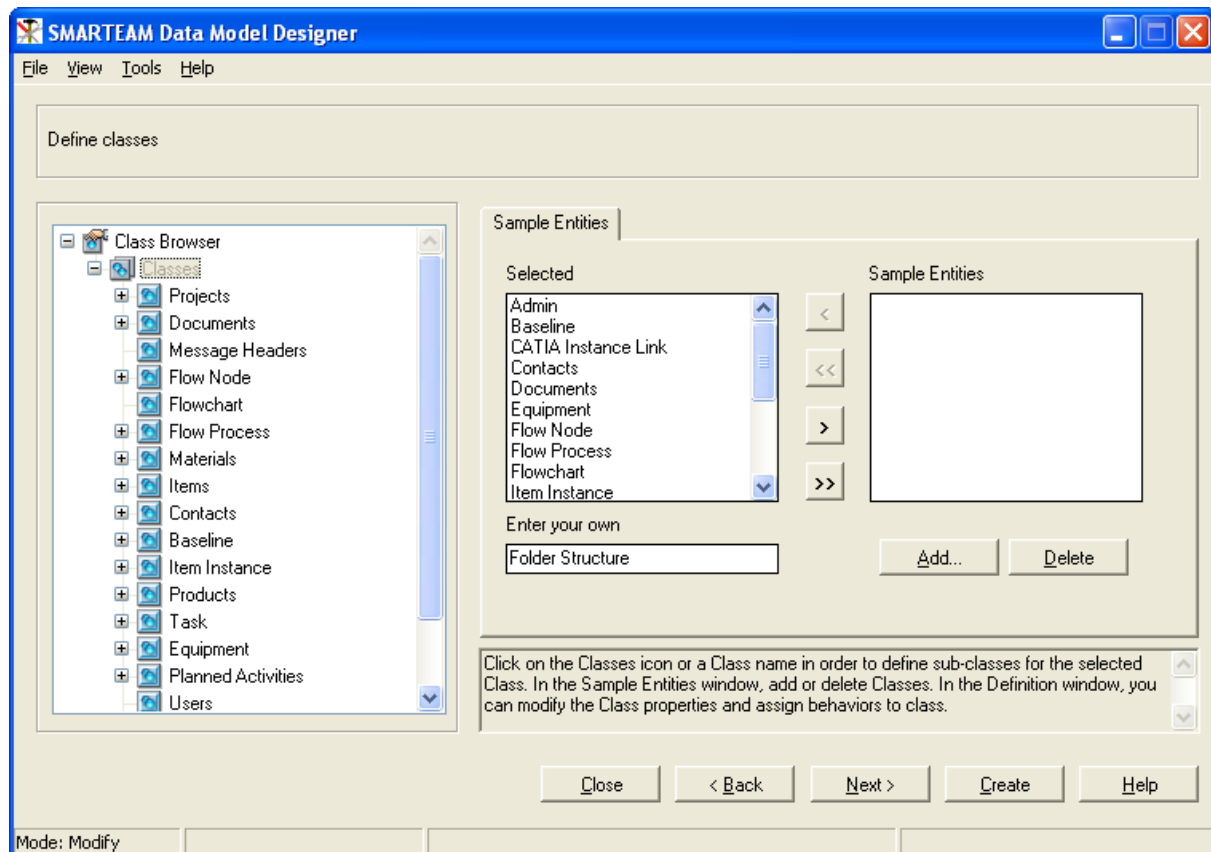
2.2.2 Set composition for Customer and contact person

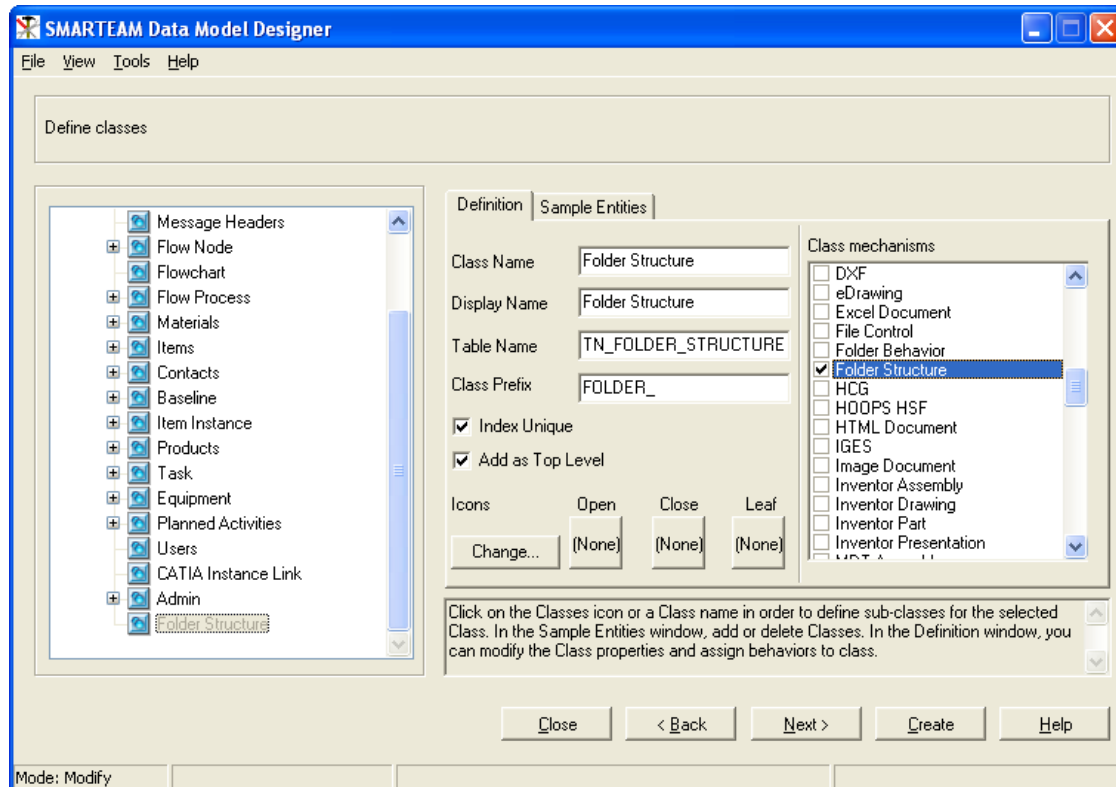
In order to display the assigned contact person to the appropriate customer you have to enable a composition between customer and the contact person. On the link definition step set a composition for the contact person



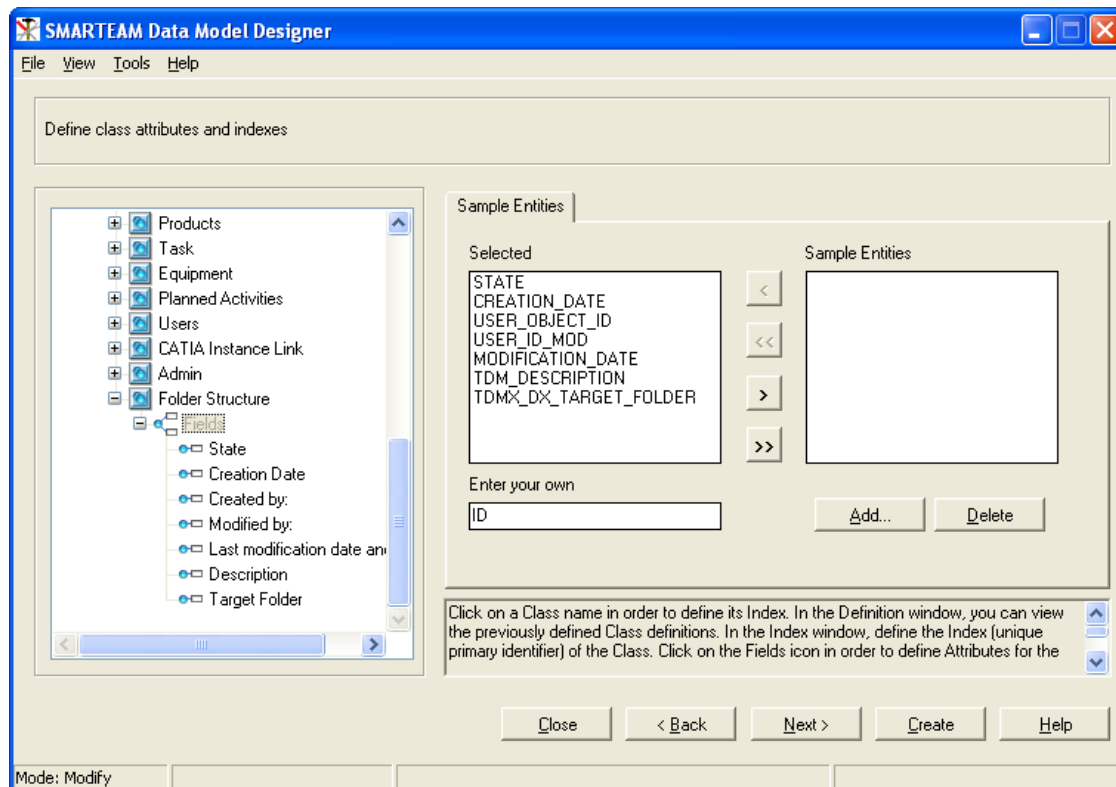
2.2.3 Enable 'Folder structure' capability

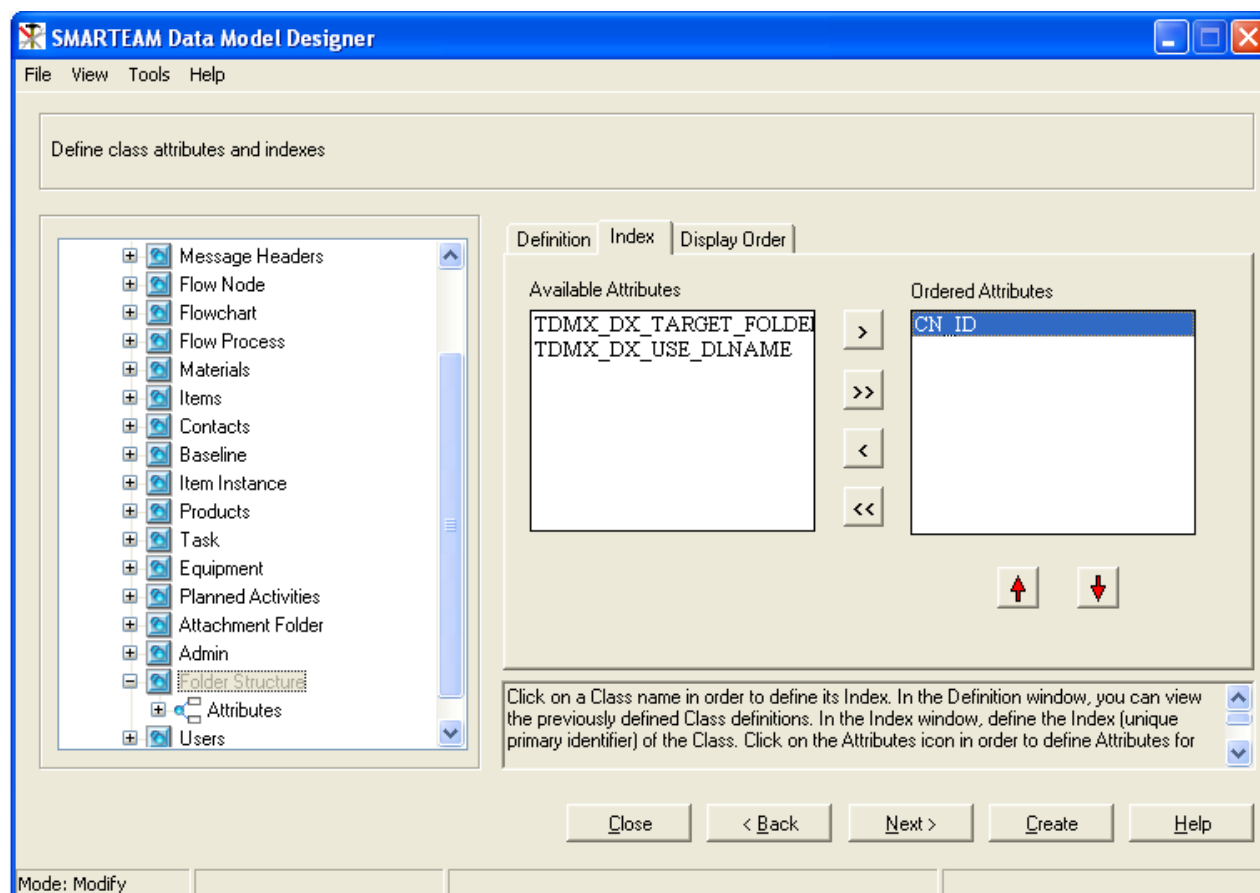
In order to enable the 'Folder structure' capability it is advised to create a new super class and check the 'Folder Structure' class Mechanism to it.





In case a new super class has been decided to create ensure that an index has been assigned to it.

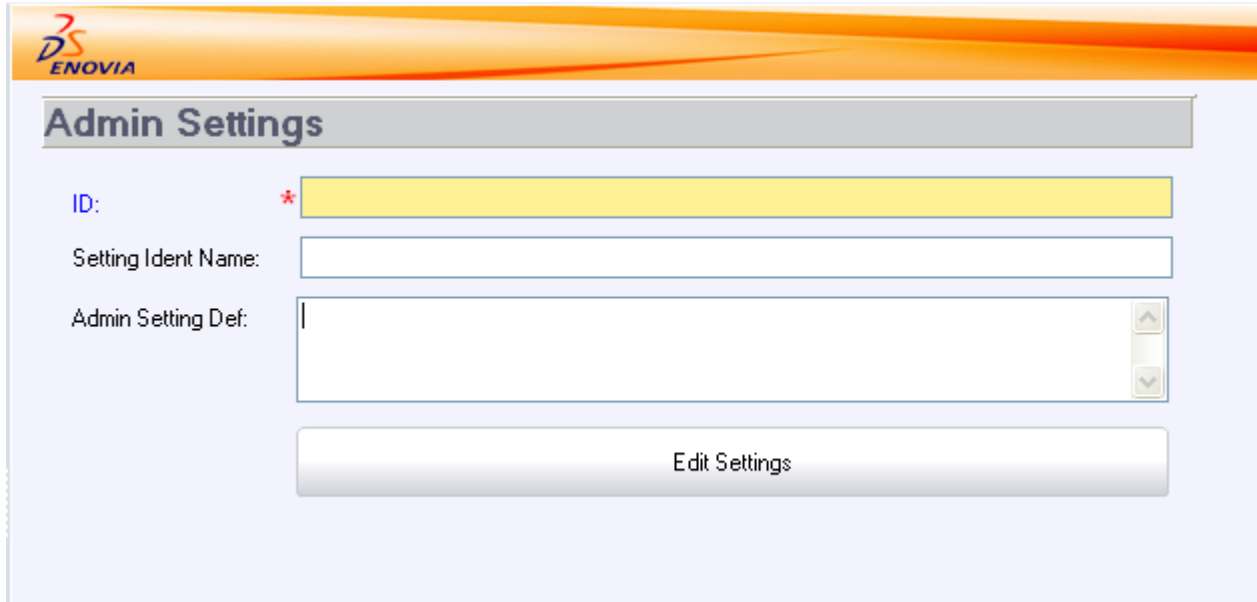




After finishing all the steps described above you can update the target database by performing the 'create' button.

2.3 Form Designer

The profile cards can be designed like the following ones:

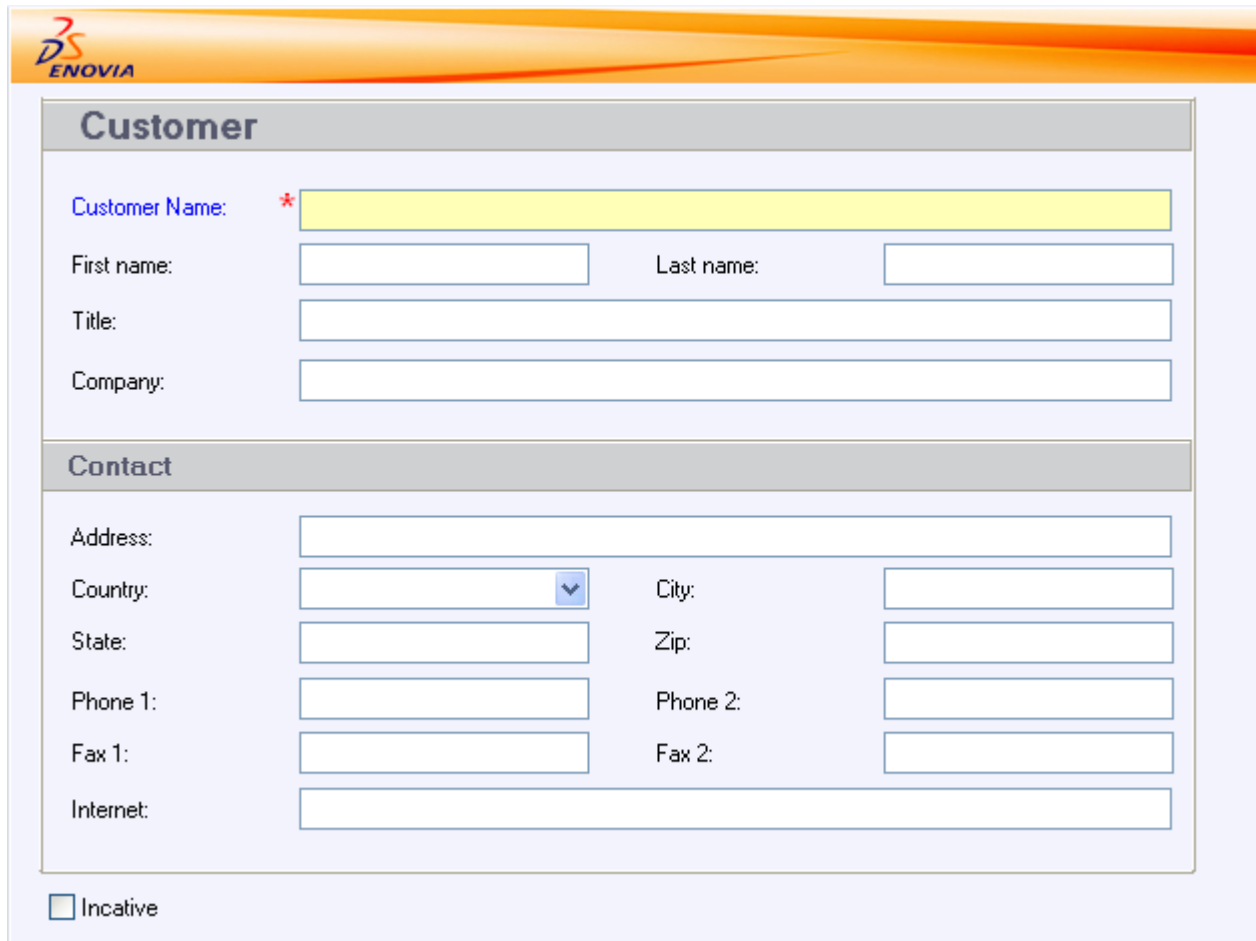


The screenshot shows a web-based form titled "Admin Settings" within the ENOVIA interface. The form includes three input fields: "ID:" (marked with a red asterisk and highlighted in yellow), "Setting Ident Name:", and "Admin Setting Def:". Below these fields is a button labeled "Edit Settings".

Figure 1: Object class Admin Settings

Setting Ident Name ≈ TDMX_VALUE1

Admin Setting Def: ≈ TDM_VALUE_LONG



The screenshot shows a web form for a 'Customer' object class. The form is divided into two main sections: 'Customer' and 'Contact'. The 'Customer' section includes fields for 'Customer Name' (marked with a red asterisk), 'First name', 'Last name', 'Title', and 'Company'. The 'Contact' section includes fields for 'Address', 'Country' (a dropdown menu), 'City', 'State', 'Zip', 'Phone 1', 'Phone 2', 'Fax 1', 'Fax 2', and 'Internet'. At the bottom of the form, there is a checkbox labeled 'Inactive'.

Customer

Customer Name: *

First name: Last name:

Title:

Company:

Contact

Address:

Country: City:

State: Zip:

Phone 1: Phone 2:

Fax 1: Fax 2:


Internet:

☐ Inactive

Figure 2: Object class Customer

Company ≈ CN_COMPANY

Inactive ≈ TDMX_DX_INACTIVE


Contact Person:

ID: *
First name: Last name:
Title:
Company:
Email address:

Contact


Address:
Country: City:
State: Zip:
Phone 1: Phone 2:
Fax 1: Fax 2:
Internet:

☐ Inactive**Figure 3: Contact Person**

First Name ≈ CN_FIRST_NAME

Last Name ≈ CN_LAST_NAME

Inactive ≈ TDMX_DX_INACTIVE

**Figure 4: Object class Data Exchange**

Process Phase ≈ TDMX_DX_PROCESS_PHASE

Exchange Direction ≈ TDMX_DX_EXCHANGE_TYPE

Status Description ≈ TDMX_DX_STATUS_INFO

Send to Company ≈ TDMX_DX_SEND_TO_COMPANY

Send to Contact Pers. ≈ TDMX_DX_SEND_TO_CONT_PERS

Send by ≈ USER_OBJECT_ID

Send at ≈ CREATION_DATE

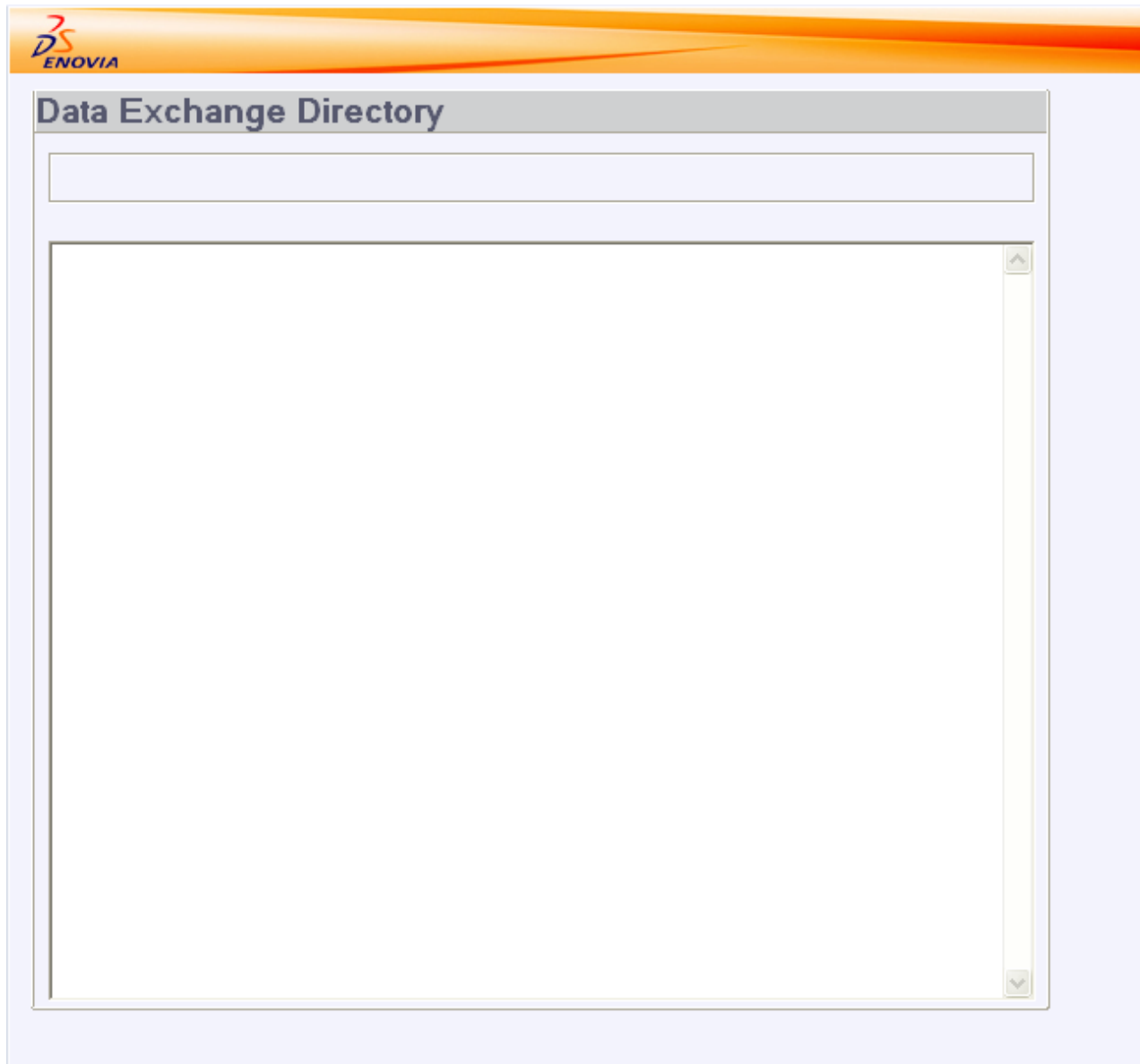


Figure 5: Object class Data Exchange

Data Exchange Directory (Hyper Link & HTML Browser) \approx TDMX_DX_EXCHANGE_ROOT_DIR

2.4 Script maintenance

To make SmartDX accessible in ENOVIA SmarTeam® you have to declare the following 'jump ins' (function name) in the Script maintenance. These function names are located in the SmartDX.ebs file. First copy this file in your script directory (mainly SmarTeam\Script):

| Function name | Operation name | Class tree level |
|---------------------------|-----------------------------|------------------|
| Admin_Tool | Admin Tool | Classes |
| SmartDX_LockObjs | Lock Object | Documents |
| SmartDX_StartExport_Struc | SmartDX Export Folder | Classes |
| SmartDX_StartExport | SmartDX Export | Classes |
| SmartDX_SelObjs | SmartDX Select Objects | Documents |
| SmartDX_DisplayDataEx_Obj | SmartDX latest Exchange Obj | Documents |
| SmartDX_UnlockObjs | Unlock Object | Documents |

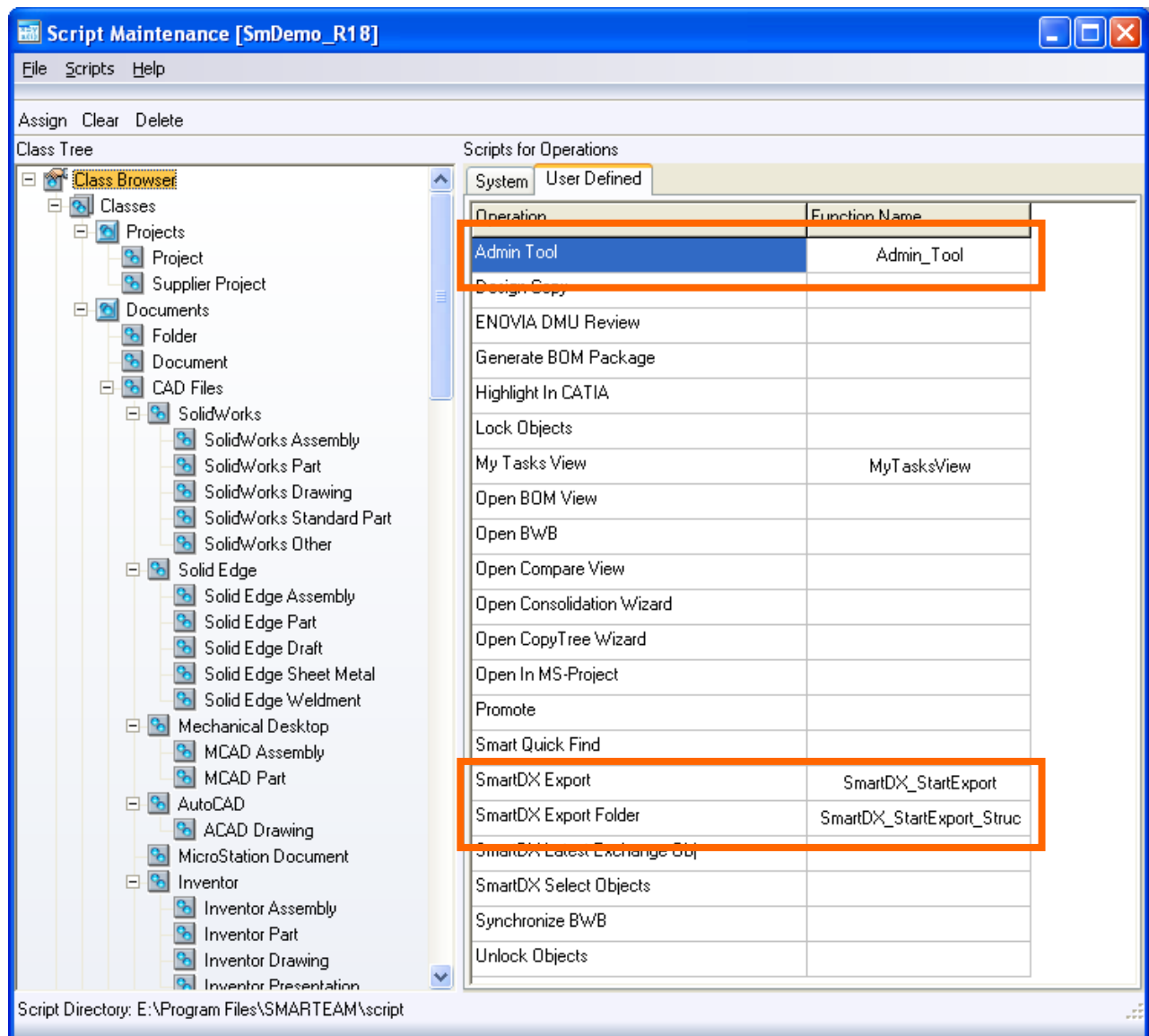


Figure 6: User defined Scripts 1

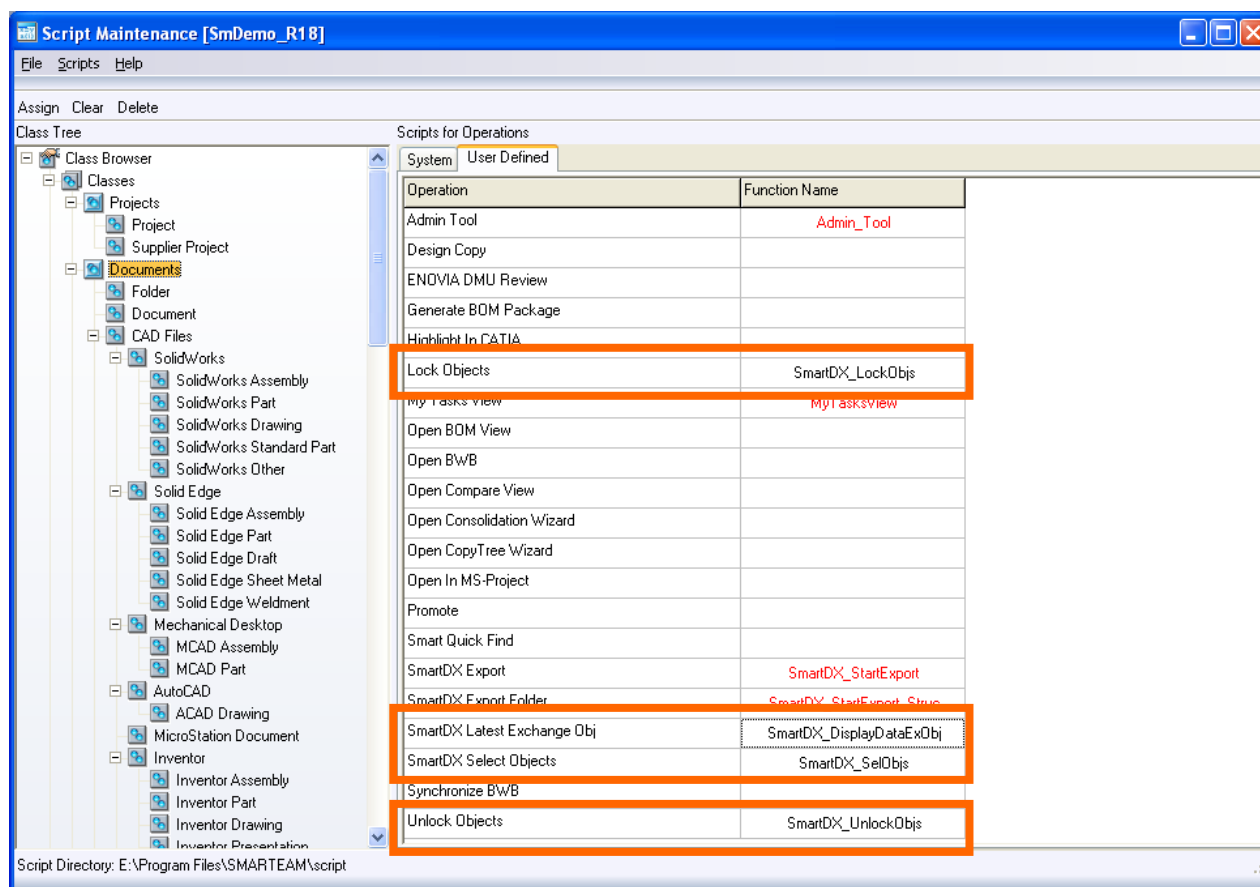


Figure 7: user defined Scripts 2

The following 'Jump Ins' (function name) have to be declared as system-scripts in the Script maintenance. These function names are located in the SmartDX.ebs file:

| Function name | Operation name | Class tree level |
|-------------------|---------------------------------|------------------|
| SmartDX_LFCStage2 | Life cycle stage 2 (after) | Documents |
| SmartDX_SetExDir | Load life cycle screen (before) | Documents |

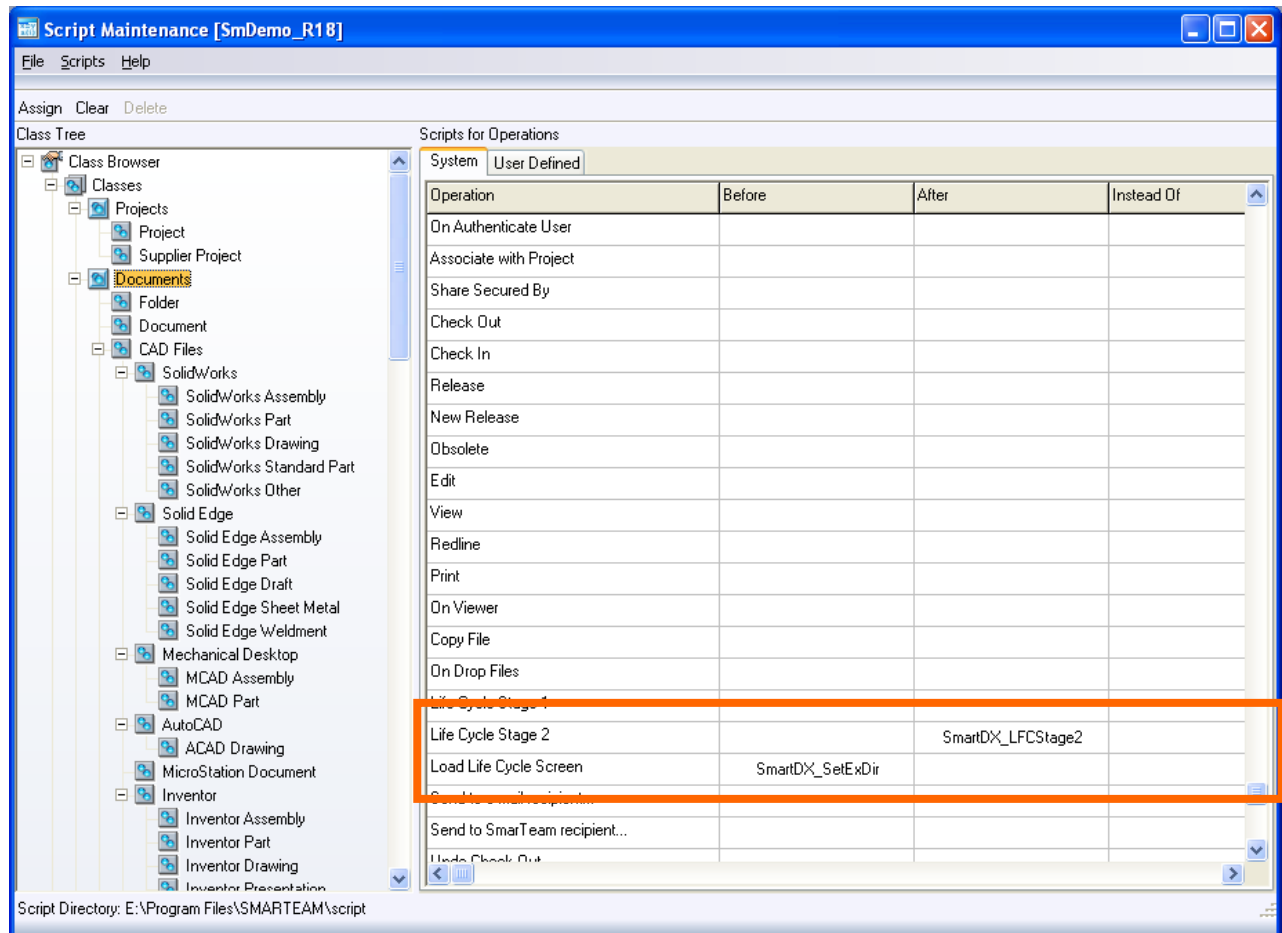


Figure 8: System scripts

2.5 Menu editor

We recommend you to use the following menu set-up

| Menu command | Operation | Menu input |
|-----------------------------|---------------------------|-------------------------------------|
| Select export objects | SmartDX_SelObjs | Contextual menu and / or Menu Icon |
| Lock object | SmartDX_LockObjs | Contextual menu and / or Menu Icon |
| Unlock object | SmartDX_UnlockObjs | Contextual menu and / or Menu Icon |
| Admin Tool | Admin_Tool | Menu icon (only for administrators) |
| SmartDX Export | SmartDX_StartExport | Menu icon |
| SmartDX Export Folder | SmartDX_StartExport Struc | Menu icon |
| SmartDX latest Exchange Obj | SmartDX_DisplayDataEx_Obj | Menu icon |

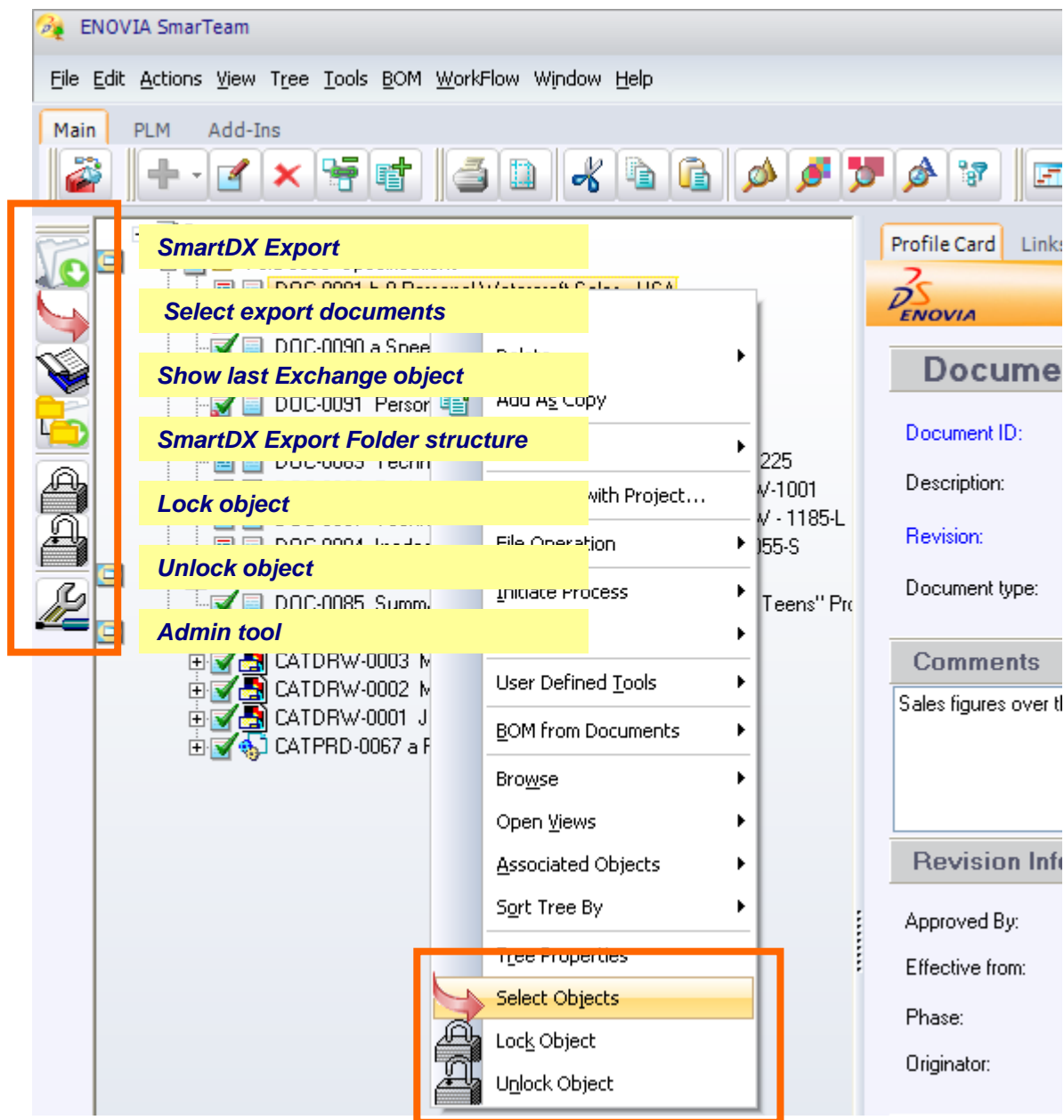


Figure 9: menu icons and contextual menus

3 Administration

3.1 3.1 Admin Tool

3.1.1 Purpose

Basically the Admin Tool defines the initial parameter to run SmartDX in a proper manner. Beside others the root exchange path and the access to the SmartALM will be defined here. The Admin Tool will be set-up and maintained using a graphical interface. Most of the parameters belong to the client site. To initialize the server site a set of common ini – files have been used instead (see chapter initialize files (server site))

3.1.2 Access privileges for the Admin Tool

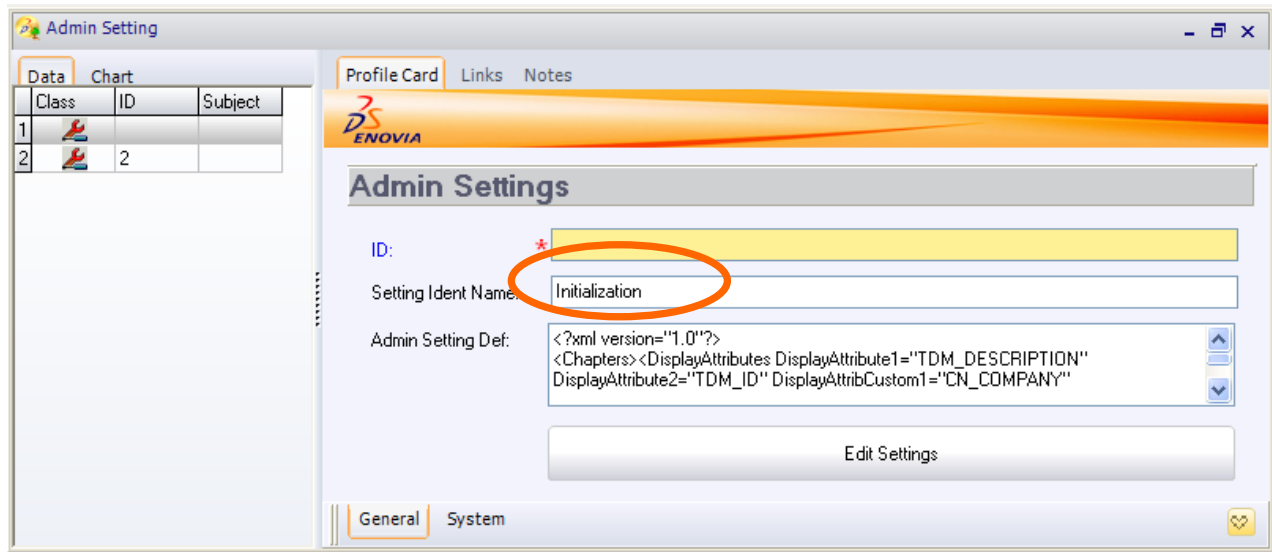
It is not necessary to have administrative access privileges to use the Admin Tool, but we recommend that only administrators should have access to it.



3.1.3 Set up the Admin Tool

The Admin Tool will be available through the 'Jump In' *Admin_Tool* from the Script *SmartDX.ebs*. It should be available using a menu icon in the workbench.


At least one AdminObject has to exist to work with the Admin Tool.

The AdminObject must have the value Initialization for the attribute *TDMX_VALUE_1* (Name).



| | Class | ID | Subject |
|---|---|----|---------|
| 1 |  | | |
| 2 |  | 2 | |

Profile Card Links Notes



Admin Settings

ID:

Setting Ident Name:

Admin Setting Def:

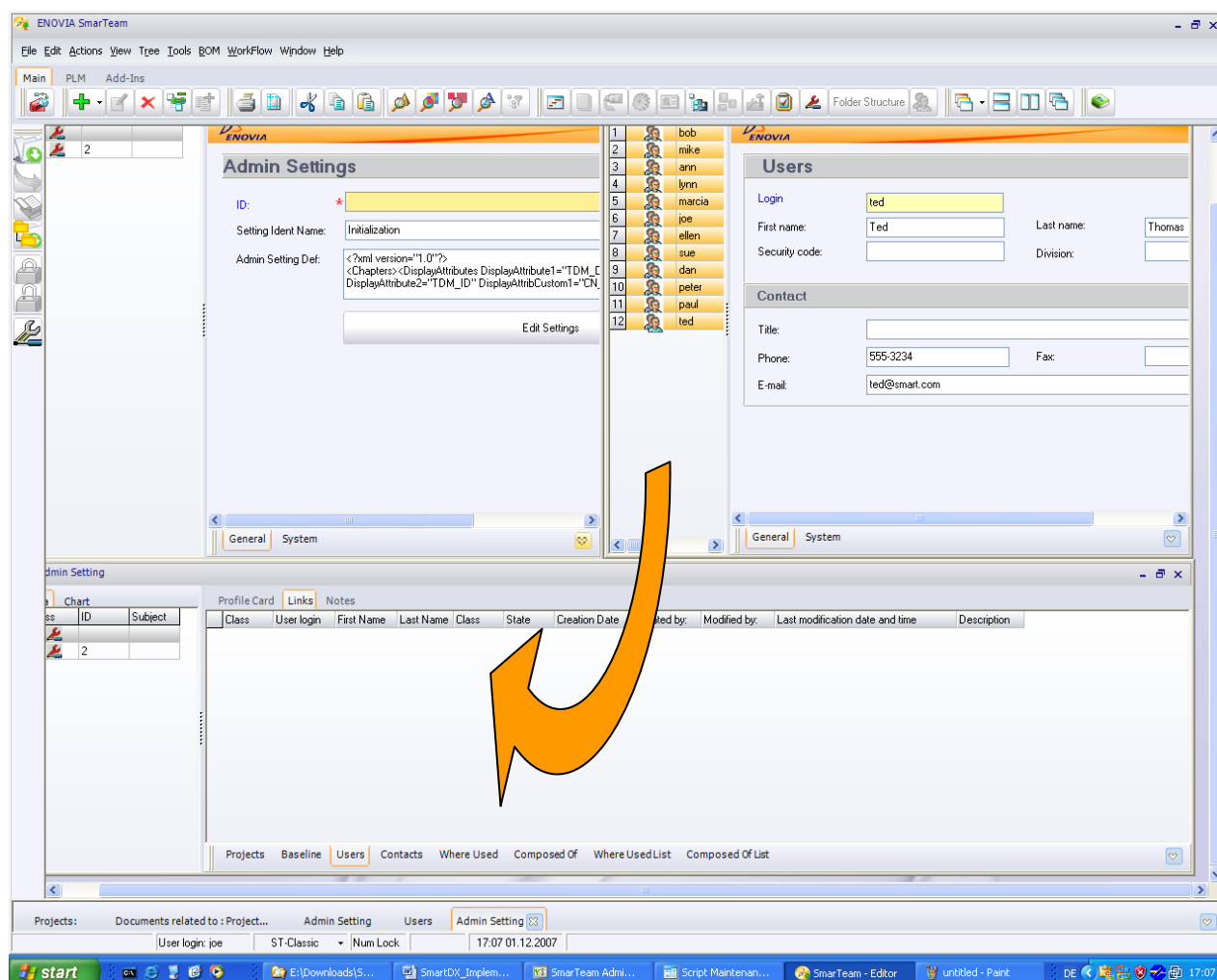
```
<?xml version="1.0"?>
<Chapters> <DisplayAttributes DisplayAttribute1="TDM_DESCRIPTION"
DisplayAttribute2="TDM_ID" DisplayAttribCustom1="CN_COMPANY"

```

Edit Settings

General System

This (or these) AdminObject(s) must have a link to all users that will work with SmartDX using these initialization parameters (i. e. different users will use different exchange directories).

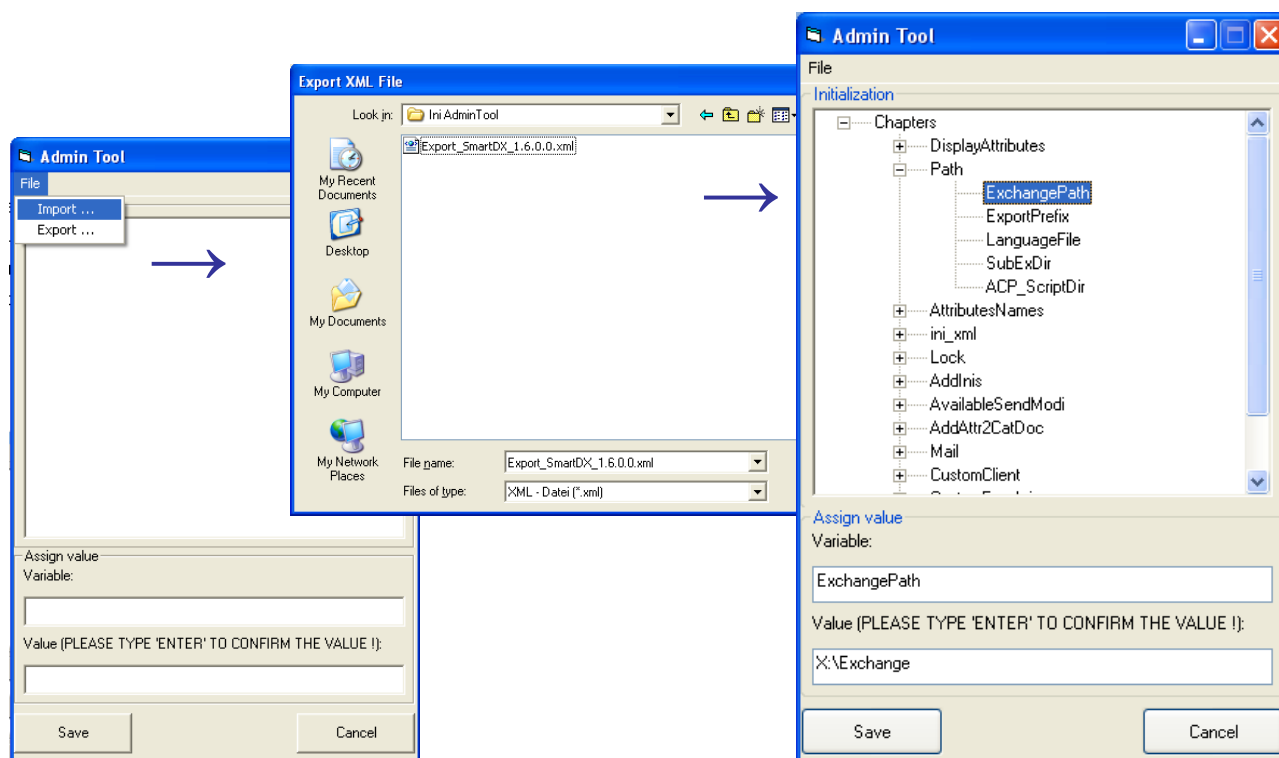


It is possible to create more than one AdminObjects to initialize SmartDX with different behaviors. Each object must have the name Initialization (TDMX_VALUE_1 Attribute). These different objects can be linked to different users. Using this you have the capability to define different exchange directories for users coming from different company localizations.

The content of the Admin Tool will be stored xml - formatted in the memo field of the appropriate AdminObject (TDMX_LONG_VALUE). When using the Admin Tool the application will read the definitions from the XML string of the memo field and represents it hierarchically structured in the Admin Tool.

To have an easy access to all the different parameters for SmartDX you can use the import function of the Admin. Inside the standard delivery there is a file called *SmartDX7.0_AdminTool.xml* (mainly ...\\Program Files\\SmartDX_Client\\Implementation Tools\\) which can be used for the first time implementation. This file

provides all necessary initialization parameters for SmartDX which can be customized user specific using the GUI of Admin Tool.



3.1.4 The Admin Tools initialization parameters

IMPORTANT NOTIFICATION:

Ensure that every SmarTeam attribute, which has been defined for use as display attribute 1 in the Admin Tool is not empty. It must have a value (i. e. check the required parameter in the SmarTeam form designer).

After importing all default parameters for SmartDX to the Admin Tool each parameter can be customized user specific. The following table explains the meaning of each parameter

| Chapter | Parameter name | Explanation | Default value |
|-------------------|-----------------------|---|-----------------|
| DisplayAttributes | DisplayAttribute1 | First display Attribute in the SmartDX wizard list for selected documents ENSURE THIS ATTRIBUTES VALUE IS NEVER EMPTY IN ENOVIA SMARTEAM® | TDM_DESCRIPTION |
| | DisplayAttribute2 | Second display Attribute in the SmartDX wizard list for selected documents. | TDM_ID |
| | DisplayAttribCustom1 | First display Attribute in the SmartDX wizard list for customers. ENSURE THIS ATTRIBUTES VALUE IS NEVER EMPTY IN ENOVIA SMARTEAM® | CN_COMPANY |
| | DisplayAttribCustom2 | Second display Attribute in the SmartDX wizard list for customers. | |
| | DisplayAttribContact1 | First display Attribute in the SmartDX wizard list for Contact person. | CN_LAST_NAME |

Implementation Guide SmartDX

| | | | |
|------|--------------------------|--|--|
| | | ENSURE THIS ATTRIBUTES VALUE IS NEVER EMPTY IN ENOVIA SMARTEAM® | |
| | DisplayAttribContact2 | Second display Attribute in the SmartDX wizard list for Contact person. | CN_FIRST_NAME |
| | DisplayAttribFolderStruc | Shows the description of the folder structure definition | TDM_DESCRIPTION |
| | DisplayAttribProj1 | Shows the first display attribute of the project that is used for project related renaming | TDM_DESCRIPTION |
| | DisplayAttribProj2 | Shows the second display attribute of the project that is used for project related renaming | TDM_ID |
| Path | ExchangePath | Root directory for data exchange. A unique named subdirectory will contain all the exchange documents. UNC path is valid. | C:\Exchange |
| | SubExDir | Name of the sub directory in the unique export directory described above | SubDir |
| | ExportPrefix | Prefix for the unique exchange subdirectory. No need to change. | Export |
| | LanguageFile | Sets the Directory name to the language files for SmartDX to translate reports, error messages and SmartDX menu captions. UNC path is valid. | C:\Program Files\SMARTEAM\SmartDX_Client\LanguageFiles |
| | ACP_ScriptDir | Location of AllCATPart Script generation | C:\Program |

Implementation Guide SmartDX

| | | | |
|-------------------|------------------|---|---|
| | | | Files\SMARTEAM\SmartDX_Server\A CPScript |
| AttributesNames | | | |
| | UserEmail | Name for the recipient email attribute (Contact person) | USER_EMAIL |
| Ini_xml | StructurFileName | Name of the xml based structure file | SmartDX_Structure.xml |
| | DataFileName | Name of the file containing the client – server transfer parameter | SmartDX_Data.xml |
| Lock | LockedInfo | Notification for locked object to display in the SmarTeam tree view. Will be written in attribute CN_LOCK_DESCRIPTION | (Object locked) |
| | UnLockRole | The role name that enables assigned user to unlock objects although they did not lock them | UnlockRole |
| AddIns | UseURL2RefExDir | Defines whether a reference, placed in the data exchange object, will be created to the exchange directory. | True |
| AvailableSendModi | Mail | Defines the display status of this transfer capability (Transfer via Email) ¹⁾ | 1 |
| | MailStructure | Like before (Transfer via Email including the structure file) ¹⁾ | 2 |
| | Odette | Like before (Server side batch routine will use the odette capability) ¹⁾ | 1 |
| | CustomSend | Like before (Server side batch routine will use | 1 |

Implementation Guide SmartDX

| | | | |
|----------------|------------|---|-----------|
| | | the custom capability) ¹⁾ | |
| | NoSend | Like before (Server side batch routine will use no transfer capability) ¹⁾ | 1 |
| AddAttr2CatDoc | CATProduct | Defines the CATIA document type to which additional attributes and / or Parameter can be added. All = Both, parameter and properties, will be mapped Property = Only properties will be mapped Parameter = Only parameter will be mapped | All |
| | CATPart | Defines the CATIA document type to which additional attributes and / or Parameter can be added. All = Both, parameter and properties, will be mapped Property = Only properties will be mapped Parameter = Only parameter will be mapped | All |
| | CATDrawing | Defines the CATIA document type to which additional attributes and / or Parameter can be added. All = Both, parameter and properties, will be mapped Property = Only properties will be mapped Parameter = Only parameter will be mapped | Parameter |

Implementation Guide SmartDX

| | | | |
|---------------|------------------------|--|-------|
| Mail | SMTPServer | Name of the SMTP server for Email transfer mode | |
| CustomClient | UsePartnerList | Defines the behavior in case of selecting the custom transfer mode whether to show the partner list in the SmartDX wizard. | True |
| | UseCustomForm | Defines the behavior in case of selecting the custom transfer mode whether to show the custom specific form for additional user input in the SmartDX wizard. | False |
| CustomFormIni | MultiLine_x | If the parameter UseCustomForm (see definition above) is set to true, you can define the amount of user inputs. Each parameter MultiLine_1 to MultiLine_n conforms to a input line in the custom form. If you set the parameter to true a multi line box will appear to the user. The caption for each input box can be defined in the language files. | False |
| Relations | UseRootObjRelation | Defines whether a link between data exchange object and the selected exchange documents will be created or not. | True |
| | UseCustomRelation | Defines whether a link between data exchange object and the selected customer / partner will be created or not | True |
| | UseContactPersRelation | Defines whether a link between data | True |

Implementation Guide SmartDX

| | | | |
|----------|-----------------------------|--|------------|
| | | exchange object and the selected contact person will be created or not | |
| Settings | GenerateStructureFile | Defines whether SmartDX generates a XML structure file including hierarchical links and dependencies or the flat structure only | True |
| | AllowMultiRecipienSelection | Display the multi recipient frame to the user during the SmartDX wizard to enable to send one set of export documents to multiple recipients | True |
| | LinkAllToMainTransObject | Defines whether to link the transmittal document to all transferred objects or only to the selected root object | true |
| | XMLStructureEncoding | Represents the country – specific XML encoding schema. | iso-8859-1 |
| | UseServerMonitor | Allows the client site user to control SmartDX Server site progress by a tray icon | True |
| | PresetCopyAllBox | By setting this value to true the checkbox in the SmartDX wizard SmarTeam Copy File window would not appear and all selected Documents will be copied at once according to the life cycle rules defined for Copy Files | False |

¹⁾ Three values can be used to define the display status:

0: The transfer capability will not be displayed

1: The transfer capability will be displayed

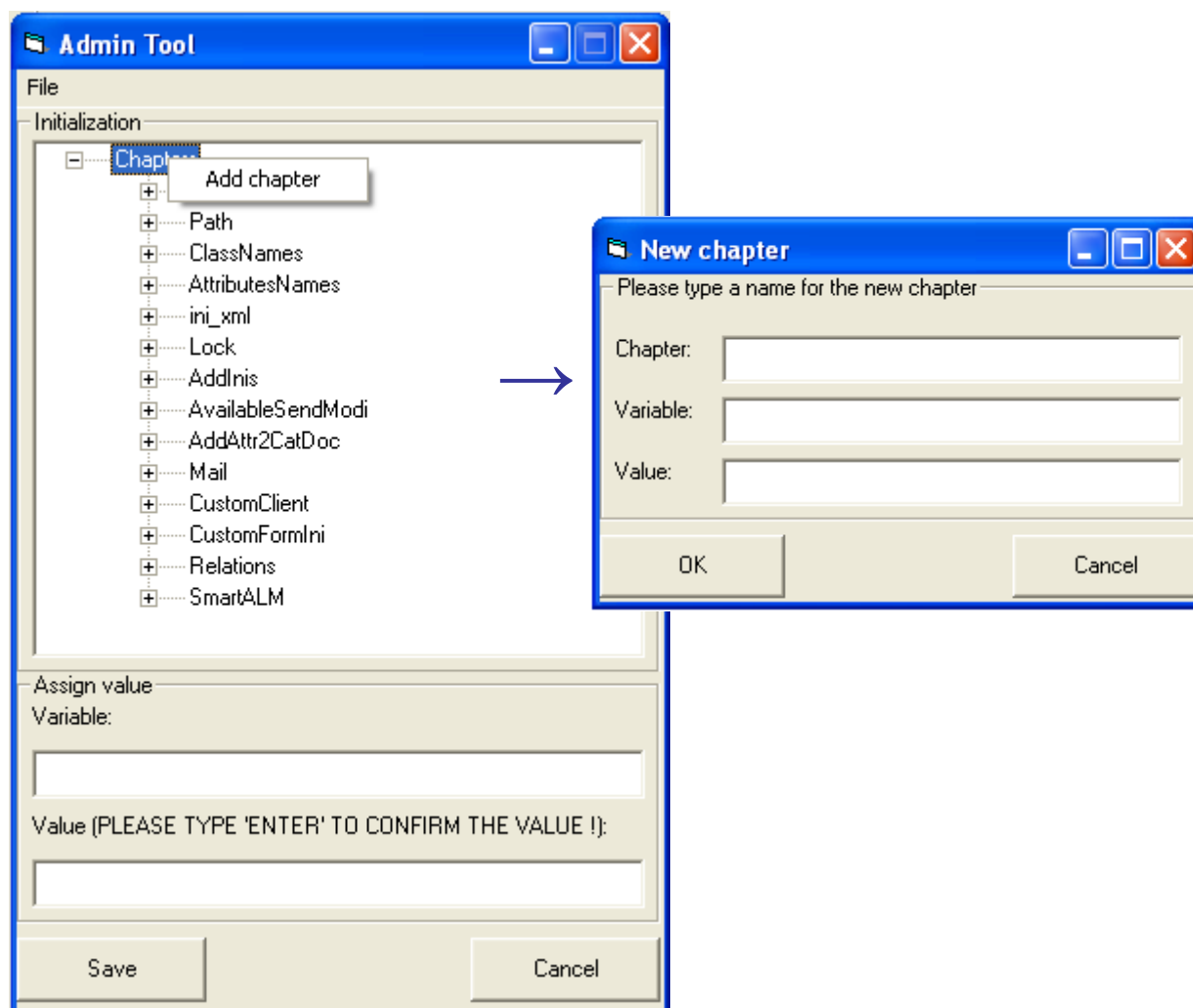
2: The transfer capability will be displayed and is selected by default

3.2 Working with the Admin Tool

3.2.1 Add chapter

In principle it is not necessary to add a chapter. However, for some update reasons it might be necessary. To add chapters use the contextual menu on the chapter node and chose *Add chapter*.

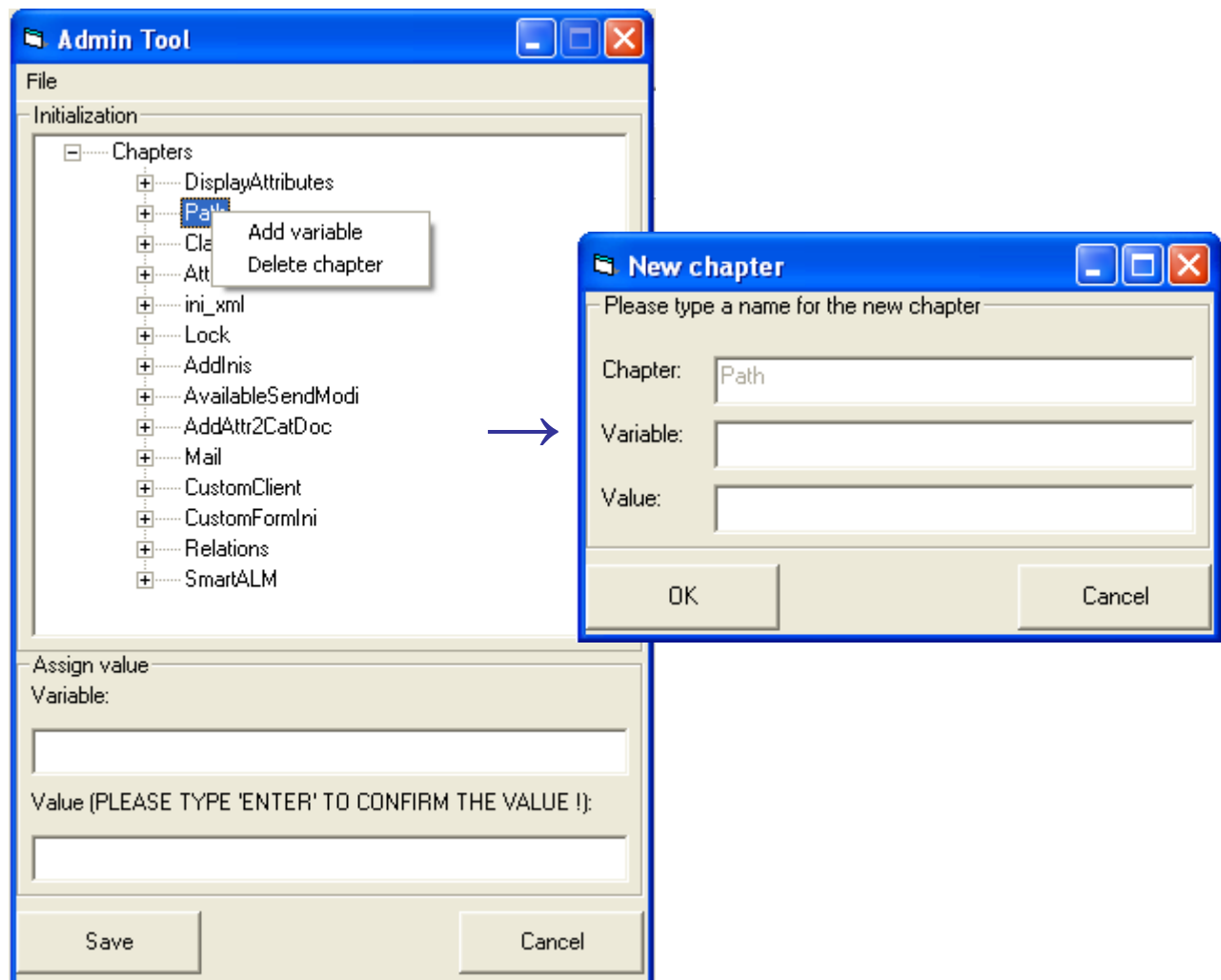
In the following window the new chapter name has to be entered. You can not use the name of an existing chapter again. Optional you can apply a new parameter and the value to the new chapter in this step.



3.2.2 Add a new parameter

You can create a new parameter using the contextual menu. Place the mouse cursor on the appropriate chapter and select Add variable.

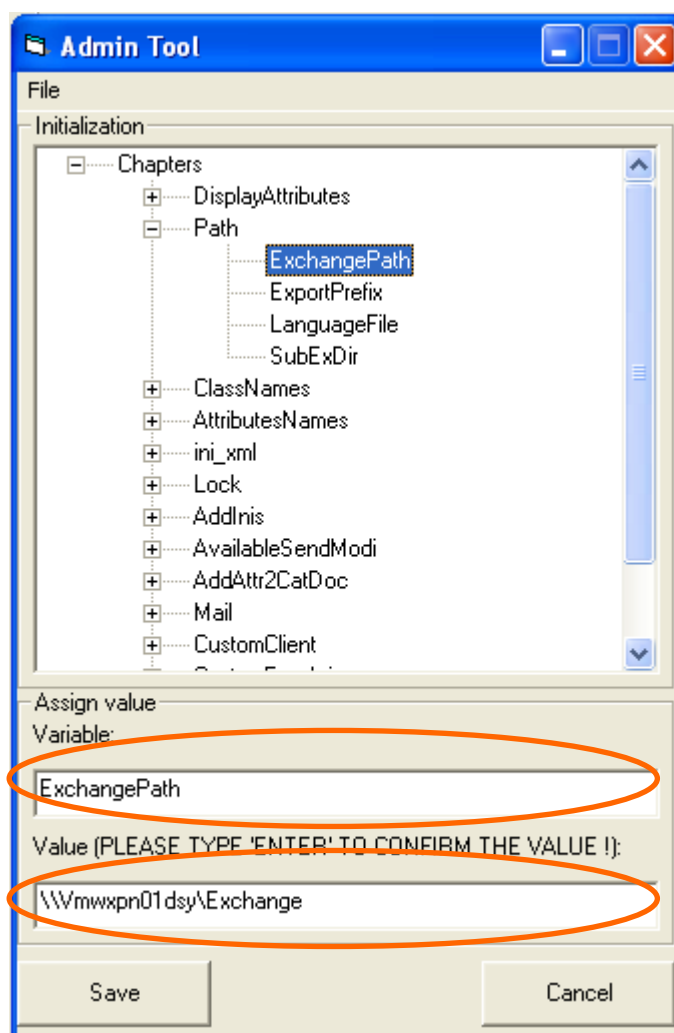
In the following window you can enter the new parameter name and optional the value for the parameter. The preselected chapter will be displayed in read only mode (light grey).



3.2.3 Change the value of a parameter

You can modify the value of a parameter after you selected it in the tree view. In the area below the tree view the current selected parameter will be displayed. The parameters value can be now overwritten in the text box below

!! Important !! The Entry in the value text box must be confirmed using the Enter key ↵



3.2.4 Save changes in the Admin Tool

To Save changes please press the Save button on the Admin Tool form. The modified content of the form will be saved xml formatted in a memo field of the AminObject.

3.3 Initialization file (Server side)

The SmartDX batch mode on the server side is initialized using standard '*.ini' files. Accordingly to the application it is located in the same directory and is named likely.

On the server side there are 3 different ini files:

- SmAdminTool.ini
- SmartDX_CallFunctions.ini
- GUIDaemon.exe.config

3.3.1 SmartDX_CallFunctions.ini

content:

```
[StartSmarTeam]
AdminUser=
AdminPassword=
[StopOnError]
CATIABatch=True
CATIARename=True
SturctureFile=True
CompressingFile=True
SendMailFile=True
SendCustomFile=True
SendOdetteFile=True
CATIAACBatch=True
[AddTransFiles]
```

```
#separate filenames by using ;
Include=SmartDX_Structure.xml ; SmartDX_Data.xml
[Settings]
DelTempFiles=True
ControlBreak=True
CATIAProcIdleTime=300
[Renaming_CATIA]
myRenaming=<TDM_DESCRIPTION>
[InvalidChars]
Invalid=* : , ; ' @ /
Substitution=_
ReplaceSpaceWithSub=True
[CATIAEnv]
myOEMEnv=CATIA.V5R20.B20
myOEMEnv_Path=C:\Documents and Settings\All Users\Application
Data\DassaultSystemes\CATEnv
myOEMEnv_StartString=C:\Program Files\Dassault
Systemes\B20\intel_a\code\bin\CNEXT.exe
[Trace]
TraceOn=false
TracePath=C:\temp
```

The chapter [StartSmarTeam] defines the administrator login for ENOVIA SmarTeam®. The AdminUser parameter represents the Login name of the administrators. The AdminPassword gives the appropriate password.

The chapter [StopOnError] defines the behavior of the SmartDX Plug Ins in case of an error. If set to true the faulty application and the server side batch will stop.

The chapter [AddTransFiles] defines the file to be included to the transmittal file collection. Each file name has to be separated by a semicolon (;).

The chapter [Settings] includes 3 parameters which defines the behavior for the deletion of the temporarily created files in the exchange directory (if set to true the temp. files will be deleted) and for the control interruption. If the last parameter ControlBreak is set to true the server side batch mode will

interrupt the exchange process and the user is able to verify the transmittal documents just before transmitting them.

The third parameter `CATIAProcIdleTime` defines the idle time in case the CATIA batch process has no progress. After this idle time the CATIA process will be interrupted and the SmartDX server site process will stop with an error.

In the chapter `[Renaming_CATIA]` you will find the associations for the CATIA renaming conventions. The notation in the customer's respectively in the contact person's object must correspond to one of the notation in the chapter `[Renaming_CATIA]`. Otherwise the Plug In including the renaming rule will run. The list for renaming conventions can be extended as long as you need.

The chapter `[InvalidChars]` indicates the invalid characters for a CATIA document name.

In the chapter `[CATIAEnv]` you will find the associations for the CATIA environment conventions. The notation in the customer's respectively in the contact person's object must correspond to one of the notation in the chapter `[CATIAEnv]`.

The chapter `[Trace]` defines the trace behavior of SmartDX server side. If the parameter `TraceOn` is set to true all operations of the batch routine will be recorded and stored in a file with application name and the suffix `.Trace`. The directory for the trace capability can be defined with parameter `TracePath`.

3.4 Language files

There are three different types of language files

1. Error messages (Error.SmDX_xx)
2. Report entries (protocol.SmDX_xx)
3. notations of the SmartDX wizard skin (buttons, menu, ...) (caption.SmDX_xx)

The layout of a language file is nearly the same. At first there is an association number.

Next to this number the entry can be replaced by the appropriate language.

The number in the curly brackets represents the current SmarTeam object.

Example:

| Language file | Output in report |
|------------------------------------|--|
| p0002=report date: {0} , user: {1} | report date: 20060404_140505 , user: joe |

With respect to the chosen SmarTeam language selection the appropriate SmartDX language file will be used. The association will be done by the SmarTeam country indicator. The extension of the language file represents this country indicator

Example:

protocol.SmDX_**de** = German

protocol.SmDX_**en** = English

3.5 Understanding the exchange root directory

The exchange path defined by the Admin Tool serves the Root directory for the data exchange. Each data exchange process generates its own unique subdirectory with the following naming convention:

| Prefix Server / Client | User name | Time stamp | State |
|------------------------|-----------|-----------------|-------|
| Export | Joe | 20060323_133048 | New |

To this unique exchange directory all the exchange documents will be copied. Also all further needed files will be generated into this directory. The uniqueness is given by the username and the timestamp.

The **prefix** can represent 2 different states

1. @@ → Client process not finished yet. Files are under work.
2. Export → Client process finished. Server side batch process is still running or has finished.

The server side process status will be represented by different **suffixes**:

1. New → Client process has finished. Server side process currently not started
2. Pending → Daemon starts the server side process Server process.
3. InWork → Server side process is in work. Several files are under work.
4. Break → Server side process has been interrupted for verification purposes (right before transmitting)
5. Finished → Server side process finished.

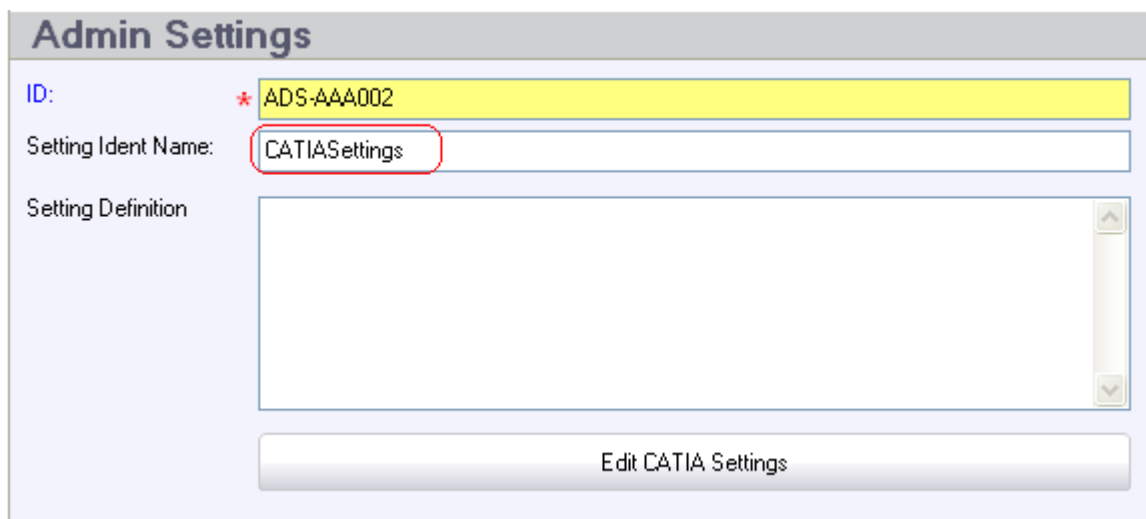
3.6 Defining the SmartDX server side behavior

To control the SmartDX server side runtime behavior a set of parameters has to be defined. According to the selected recipient these set of parameters then will drive the CATIA® batch routine of the SmartDX server. This enables a customer respectively contact person specific conversion of the CATIA® documents.

To use the Admin tool look & Feel to define the Catia specific settings you can place a button on the Admin Settings profile card and attach the script SmartDX.ebs → 'Settings_tool' to the OnClick Event.

These parameters are stored in an 'Admin Setting' Object that can be linked to the appropriate customer and / or contact person.

The Identification name for the 'Admin Setting' Object has to be **CATIASettings**.



The image shows a screenshot of the 'Admin Settings' dialog box. It has a title bar 'Admin Settings'. Inside, there are three fields: 'ID:' with a red star icon and the value 'ADS-AAA002' (highlighted in yellow), 'Setting Ident Name:' with the value 'CATIASettings' (circled in red), and 'Setting Definition' which is an empty text area. At the bottom, there is a button labeled 'Edit CATIA Settings'.

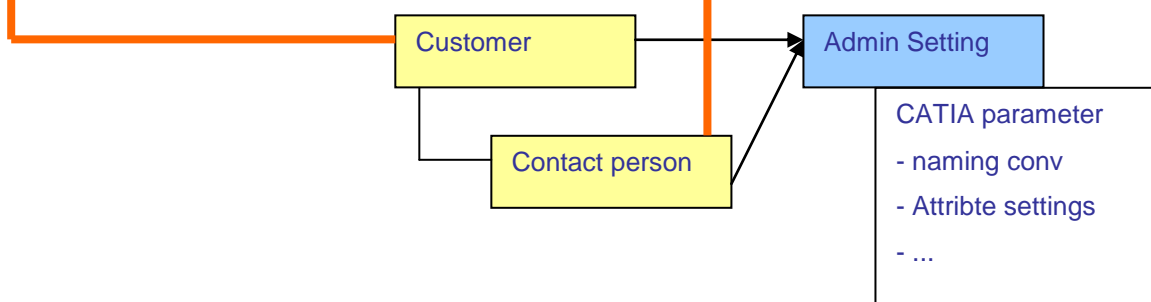
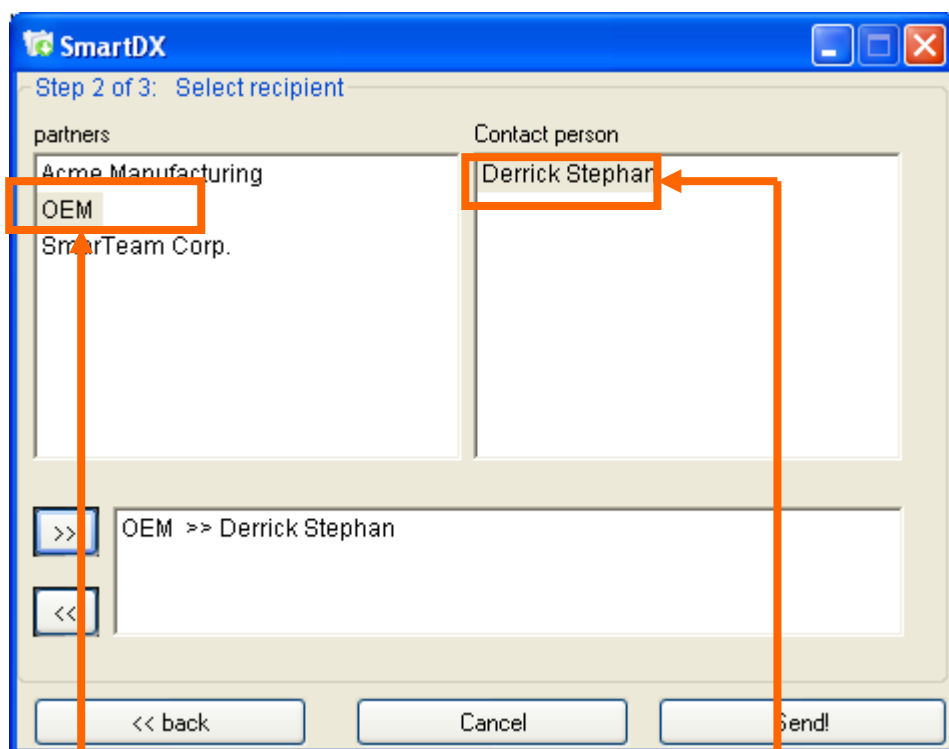
During the SmartDX wizard the user is prompted to select a recipient comprising of a customer and a contact person.

Customer and contact person can be linked to different 'Admin Setting' objects. In case of different parameter definition the one of the contact person will have the higher priority so that this parameter will be taking into account.

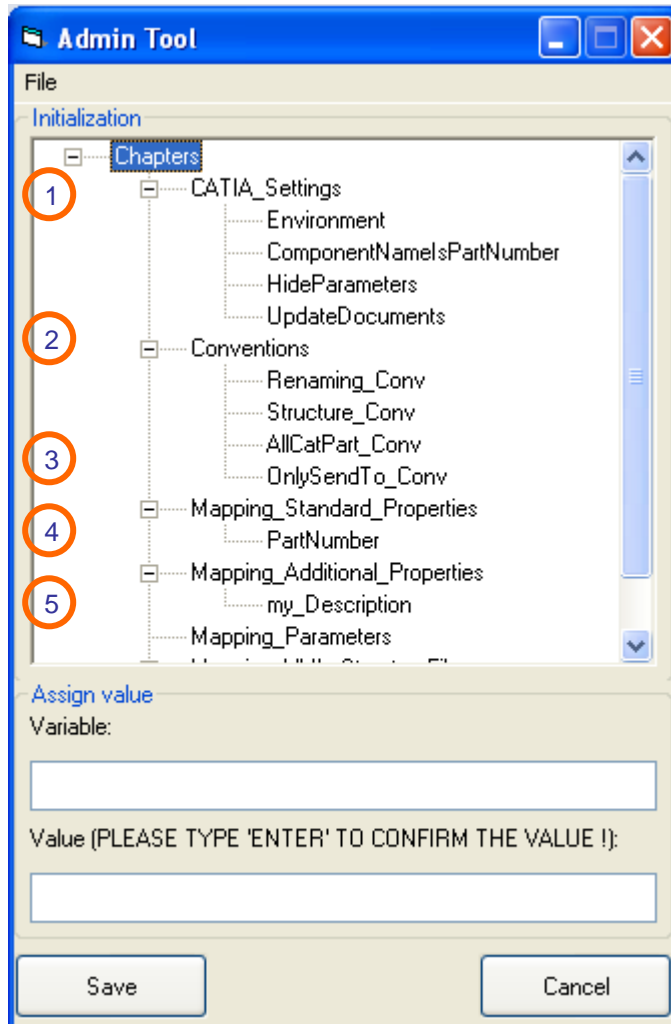
In case of blank parameter definitions the action according to these parameters will not be performed.

The following example describes the parameter priority.

| 'Admin Setting' Obj. linked to Customer object | 'Admin Setting' Obj. linked to Contact person object | Validity |
|---|---|---------------|
| Test_Standard | | Test_Standard |
| Test_Standard | Test_new | Test_new |
| | Test_new | Test_new |
| | | No action |
| | | |



3.6.1.1 'Admin Setting' Objekt



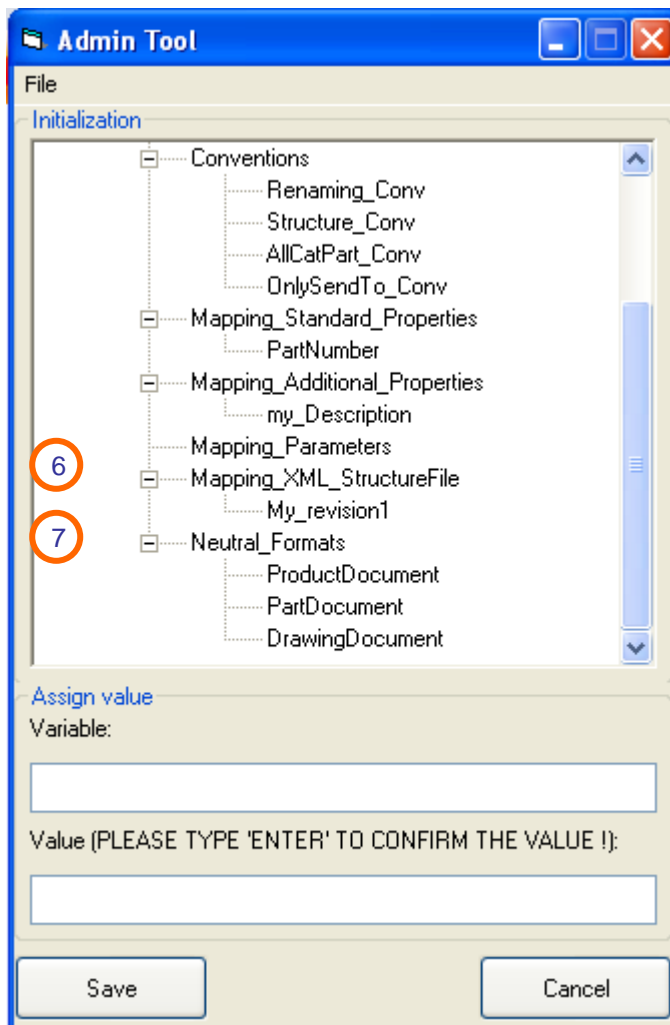
1. CATIA_Settings

- a. *Environment*: Name of the CATIA environment setting corresponding to the parameter in the ini file on the server.
- b. *ComponentNameIsPartNumber*: If this parameter is set to true the component name of the CATIA document will be equal to the PartNumber of the same CATIA document.
- c. *HideParameters*: In case additional parameters shall be set to the CATIA document this indicates whether to hide them (true) or not (false)
- d. *UpdateDocuments*: This indicates whether to update the CATIA documents after converting or not

2. Conventions

- a. *Renamin_Conv*: Name of the renaming convention for CATIA documents

- b. *Structure_Conv*: Name for the additional structure file convention which could be generated based on the default xml structure file during SmartDX server side process using an appropriate Plug In.
 - c. *AllCatPart_Conv*: Indicates whether to generate a so called AllCatPart during the CATIA batch routine or not (New in Drop 6). The AllCatPart will be generated only from the selected CATIA root documents. The name will be the name after conversion plus '_AllCATPart'
 - d. *OnlySendTo_Conv*: Indicates whether to perform the CATIA SendTo functionality to the selected documents (Not realized in this drop).
3. *Mapping_Standard_Properties*: Name of the CATIA property fields and the SmarTeam attribute field. The content of the SmarTeam attribute fields will be mapped to the default CATIA properties.
Variable: Name of the CATIA standard Property
Value: Name of the SmarTeam attribute that will be mapped
4. *Mapping_Additional_Properties*: Name of the additional CATIA property fields and the SmarTeam attribute fields. The content of the SmarTeam attribute fields will be mapped to the additional CATIA properties.
Variable: Name for the additional CATIA Property
Value: Name of the SmarTeam attribute that will be mapped
5. *Mapping_Parameters*: Name of the additional CATIA Parameter fields and the SmarTeam attribute fields. The content of the SmarTeam attribute fields will be mapped to the additional CATIA parameter.
Variable: Name for the additional CATIA Parameter
Value: Name of the SmarTeam attribute that will be mapped



6. *Mapping_XML_StructureFile*: Additional Attributes to be mapped from SmarTeam attributes to the xml based structure file (SmartDX_Structure.xml) as additional metadata information
Variable: Name for the additional Structure file attribute (SEE CHAPTER 3.13.1 FOR NAMING RULES)
Value: Name of the SmarTeam attribute that will be mapped
7. *Neutral_Formats*: Set the neutral formats (i. e. iges, step, stl, dxf, etc.) the selected CATIA documents will be converted to automatically during the SmartDX server side process
 - a. *ProductDocument*: The neutral formats available and licensed in CATIA can be set for product documents
 - b. *PartDocument*: The neutral formats available and licensed in CATIA can be set for part documents
 - c. *DrawingDocument*: The neutral formats available and licensed in CATIA can be set for drawing documents

3.7 Set up the CATIA environment for SmartDX server side batch process

To run CATIA on the server in a batch mode i. e. to rename the documents in the appropriate environment corresponding to the customer respectively contact person settings this environment has to be set up in the SmartDX_CallFunctions.ini file. The chapter [CATIAEnv] contains three parameters to start CATIA in the correct environment. These three parameters results in a CATIA start string whereby a customer individual CATIA environment can be set for the CATIA documents converting.

1. The customer respectively the contact person contains the environment setting name
2. The same name has to be taken for the three parameters in the SmartDX_CallFunctions.ini file
3. The value for the first parameter is the name of the CATIA environment text file
4. The second parameter owns the same parameter name followed by a suffix *_Path*. The assigned value contains the CATIA environment directory.
5. The third parameter owns the same parameter name as the first one followed by a suffix *_StartString*. The assigned value contains the CATIA start string.

| Attribute value | Parameter in Ini - file | example |
|--|-------------------------------|--|
| ENOVIA SmarTeam® customer obj | | |
| Test_Standard_R20 | Test_Standard_R20 | CATIA.V5R20.B20 |
| | Test_Standard_R20_Path | C:\CATIAEnvironments |
| | Test_Standard_R20_StartString | C:\Programe files\Dassault Systemes\B20\intel_a\code\bin\CNEXT.exe |

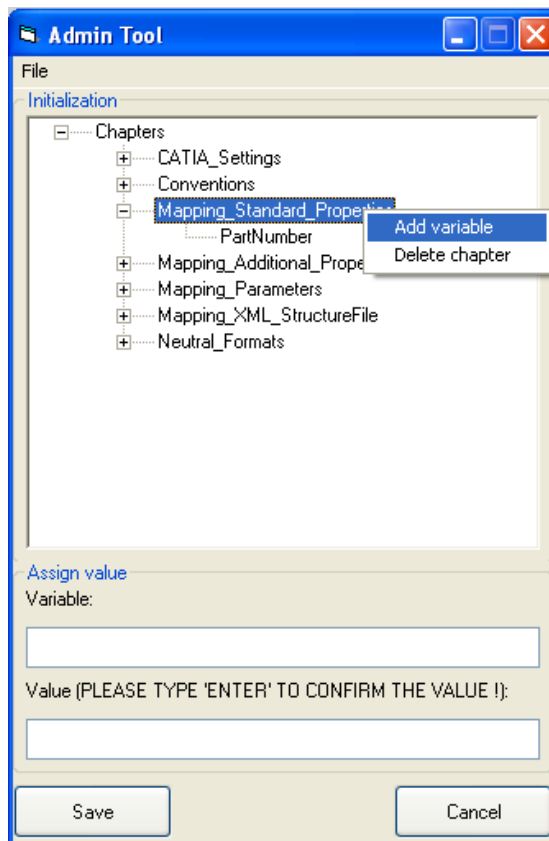
2 examples for an ini – file:

```
[CATIAEnv]
Test_Standard_R20=CATIA.V5R20.B20
Test_Standard_R20_Path=C:\CATIAEnvs\DassaultSystemes\CATEnv
Test_Standard_R20_StartString=C:\Programme\Dassault
Systemes\B20\intel_a\code\bin\CNEXT.exe
Test_R16=CATIA.V5R16.B16
Test_R16_Path= C:\CATIAEnvs\DassaultSystemes\CATEnv
Test_R16_StartString=C:\Programe files\Dassault Systemes\B16\intel_a\code\bin\CNEXT.exe
```

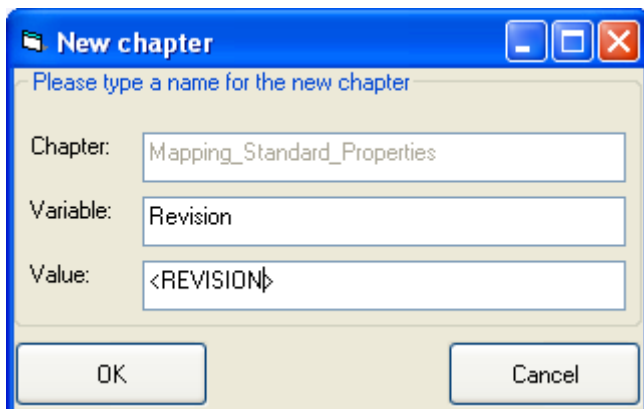
3.8 Set up CATIA Standard Attributes

To use the capability to assign additional metadata information coming from SmarTeam attributes in the CATIA properties you have to do the following steps:

1. In the 'Admin setting' GUI create a new variable in context to the chapter 'Mapping_Standard_Properties'

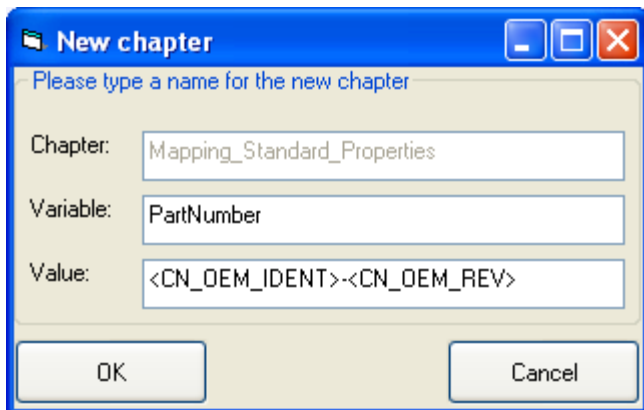


2. Define the standard CATIA property name for the value field and the ENOVIA SmarTeam® attribute name that shall be used for mapping purposes in sharp brackets



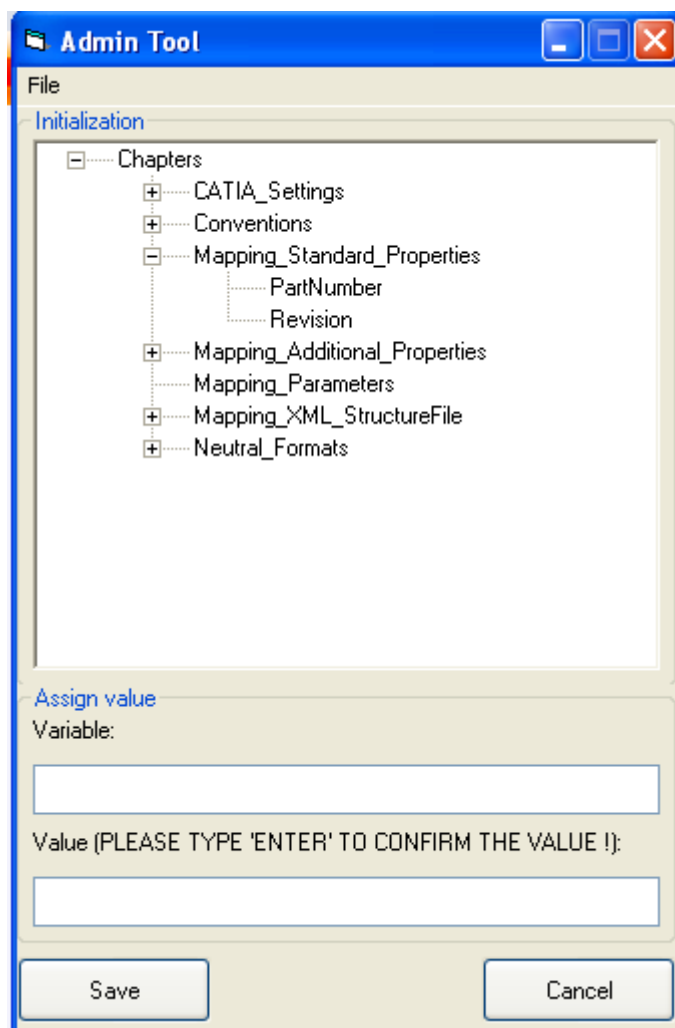
A screenshot of a Windows-style dialog box titled "New chapter". It has a blue title bar with standard minimize, maximize, and close buttons. The main area is light beige and contains the text "Please type a name for the new chapter". Below this are three input fields: "Chapter:" with the text "Mapping_Standard_Properties", "Variable:" with the text "Revision", and "Value:" with the text "<REVISION>". At the bottom are two buttons: "OK" and "Cancel".

You can also use multiple SmartTeam attributes to fill the CATIA property like



A screenshot of a Windows-style dialog box titled "New chapter", identical in layout to the one above. It contains the text "Please type a name for the new chapter". The input fields are: "Chapter:" with "Mapping_Standard_Properties", "Variable:" with "PartNumber", and "Value:" with "<CN_OEM_IDENT><CN_OEM_REV>". The "OK" and "Cancel" buttons are at the bottom.

3. The result should look like this



The following CATIA standard properties can be used for mapping:

Nomenclature
 Definition
 PartNumber
 DescriptionRef
 Source
 Revision

Assign a SmarTeam LookUp table to the CATIA property ,source'.

3.9 Set up CATIA additional attributes

The strategy to set up a new CATIA additional attribute is still the same described in chapter 3.8.

The only difference is the purpose of the variable name. In this case the variable name is the name for the additional CATIA attribute

3.10 Set up CATIA additional parameter

The strategy to set up a new CATIA additional parameter is still the same described in chapter 3.8.

The only difference is the purpose of the variable name. In this case the variable name is the name for the additional CATIA parameter.

3.11 Set up CATIA renaming rules

A renaming of the CATIA documents will be performed on the SmartDX server side application.

There are two capabilities to set up a renaming rule.

3.11.1 Renaming through concatenating SmarTeam attributes

This is the easiest way to set up a renaming rule for CATIA documents. The rule name has to be established in the appropriate parameter of the Admin Setting object. This rule name has to be entered in the SmartDX_CallFunctions.ini as well under the chapter [Renaming_CATIA]. The server side applies the defined rule to all CATIA documents related to the data exchange process. The rule underlies the following convention:

rulename=<attribute_name_1>_<attribute_name_2>_<....

The SmarTeam attribute name has to be enclosed in sharp brackets. The separator '_' can be replaced by any separator you like to use.

All blanks and invalid characters will be replaced by a character you have to specify in the

SmartDX_CallFunctions.ini under chapter [InvalidChars]. In the parameter Invalid you are able to define a set of invalid characters for a CATIA name separated by a blank (i. e. Invalid=* : , ; ' @ /). The substitution will be defined in the next parameter Substitution (i. e. Substitution=_). The next parameter defines whether you want to replace blanks in CATIA names by the same substitution (i. e. ReplaceSpaceWithSub=True).

!! Important note !! Ensure for each SmarTeam attribute you have defined for renaming has a value unequal to blank. Otherwise the server side process runs into an error due to save limitations. It is not possible to save a document like '.CATPart'.

If a SmarTeam attribute defined for renaming has no value it will be ignored for renaming. In case every defined attribute has no value this will result in an error of the batch mode.

For example you defined the rule Test=<TDM_DESCRIPTION> and for one of the CATIA objects the attribute TDM_DESCRIPTION has no value the return value for the new CATIA name would be '.CATPart'. Due to limitations in saving those files the server side application will stop the process.

Special rules for renaming:

| Notation | Description |
|--------------------------|---|
| <Date> | The current date using the format yyyyymmdd will be inserted in the new file name. |
| <Time> | The current time using the format hhmmss will be inserted in the new file name. |
| <Date+Time> | The current date and time using the format yyyyymmdd_hhmmss will be inserted in the new file name. |
| <RefToClass.Attribuname> | In case the SmarTeam attribute is type of a 'reference to class' one can declare the attribute name of the referenced object to receive the appropriate value. Both attributes has to be separated by a dot. The value will be inserted in the new file name. <TDM_APPROVED_BY.FIRST_NAME> provides the first name of the person who approves the object. |
| <LookUp Table> | In case the SmarTeam attribute is type of a LookUp table the description of the selected LookUp table value will be inserted in the new file name. |

Example 1:

Parameters in SmarTeam Admin Setting object:

Renaming Conv:

TestRenaming

Parameter in Ini – file:

```
[Renaming_CATIA]
TestRenaming=<TDM_ID>_<TDM_DESCRIPTION>
```

| Origin name | Renamed name |
|---------------|--------------------------------|
| Klotz.CATPart | CPRT-AAA013_KlotzGross.CATPart |

Example 2:

Parameters in SmarTeam Admin Setting object:

Renaming Conv:

TestComplex

Parameter in Ini – file:

[Renaming_CATIA]

TestComplex=<Date>_<TDM_APPROVED_BY.FIRST_NAME>_<TDM_APPROVED_BY.LAST_NAME>

| Origin name | Renamed name |
|---------------|----------------------------|
| Klotz.CATPart | 20060531_Joe_Smith.CATPart |

3.11.2 Individual defined renaming rules

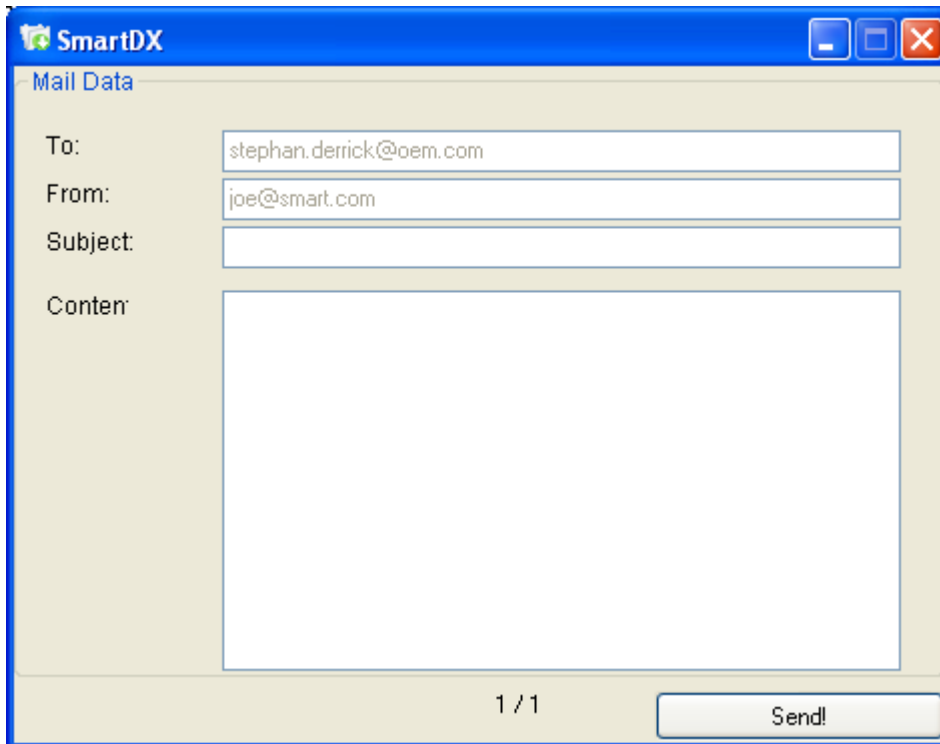
If the capability described before does not fit the requirements the second one can be used to define rename rules. This capability provides a renaming library. This library will be delivered with source code so one is able to fulfill the customer needs by customizing this library.

If the name of the renaming rule does not match any parameter in the SmartDX_CallFunctions.ini then the renaming library will run. The declared name for the renaming rule should match a functions name in the library.

For more detail please refer to the 'CatiaRename_Proj.vbp' and start it in Visual Basic 6. This project includes a detailed example on how to create your own customized renaming rule.

3.12 Setup Mail transfer

Assumed the user selected the mail transfer mode in the first window of the SmartDX client side wizard a mail attached with the data exchange files will be created automatically. In the client side GUI a window will appear to the user to type the subject and the content of the mail. This mail then will be send using the defined SMTP server.

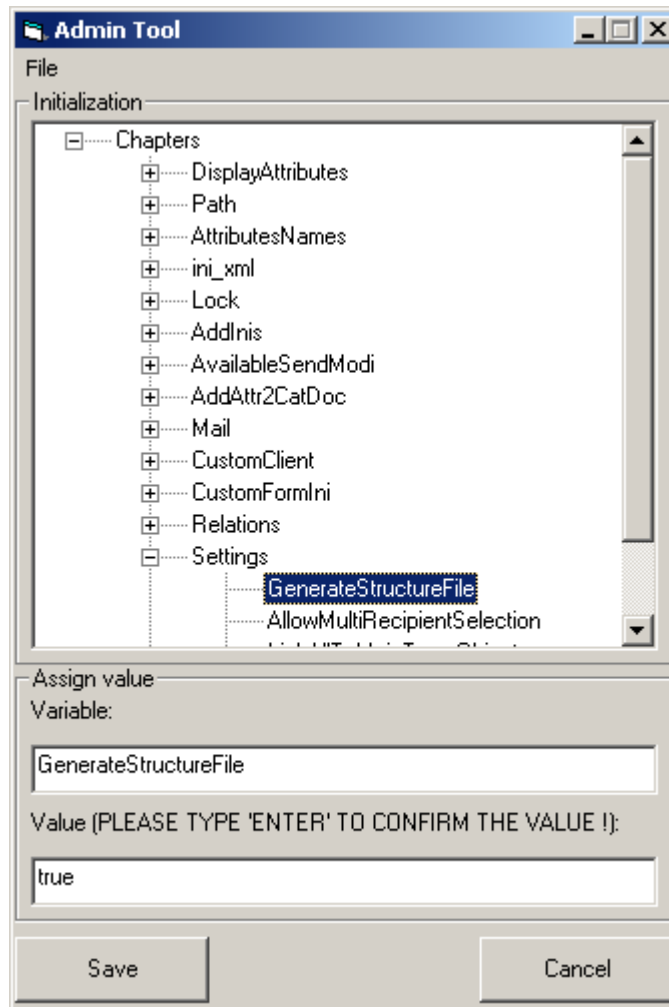


The image shows a Windows-style dialog box titled "SmartDX". Inside the dialog, there is a section labeled "Mail Data". This section contains four input fields: "To:" with the value "stephan.derrick@oem.com", "From:" with the value "joe@smart.com", "Subject:" which is empty, and "Content:" which is a large empty text area. At the bottom of the dialog, there is a status bar showing "1 / 1" and a "Send!" button.

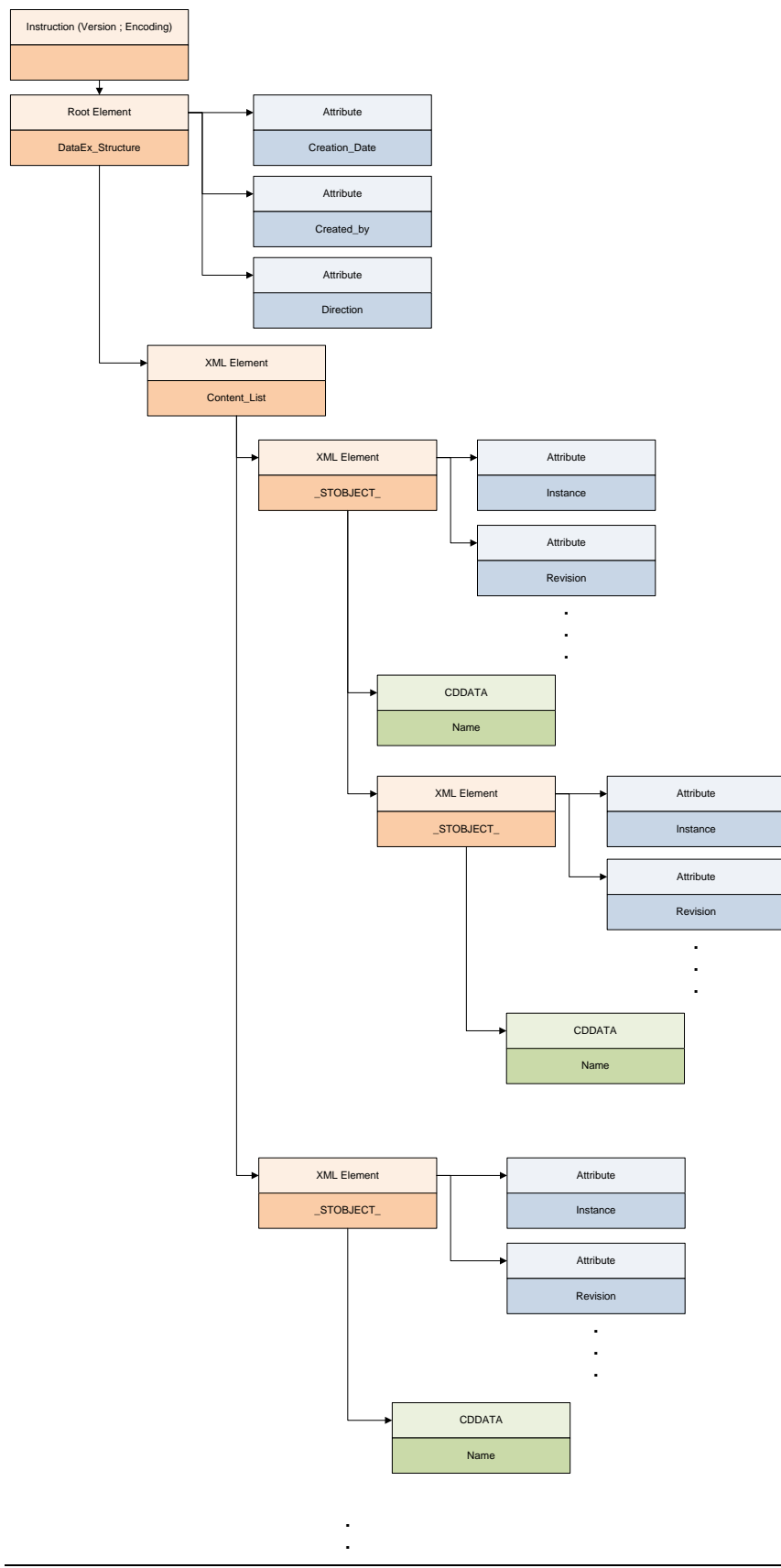
The source code for the library *SmartDX_SendMail.dll* to send mails via SmartDX will be delivered with the SmartDX installation. The source code can be customized to fit the specific customers need. The source code can be used as an example on how to customize the other transfer mode capabilities like Odette or the custom send mode.

3.13 XML Structure File

The XML structure file contains all information about the export documents. Optional you can have beside the content list a hierarchical view and a dependency list. The option can be set in the Admin tool (to set this option can raise performance problems for large CATIA assemblies!).



In the following diagram you will find the schema of the XML structure file. This is useful in case you intend to read, enhance or convert the XML file (conversion can be done in the server side Plug In 'StructureFile').



3.13.1 XML naming rules for additional SmarTeam attributes

You have the capability to add attributes in the XML structure file per each SmarTeam object you intend to export (See chapter 3.6.1.1 for details). To give a well name you have to follow the following rules:

XML elements must follow these naming rules:

- Names can contain letters, numbers, and other characters from A-Z, a-z and 0-9
- Names cannot start with a number or punctuation character
- Names cannot start with the letters xml (or XML, or Xml, etc)
- Names cannot contain spaces

Any name can be used, no words are reserved.

3.14 Setup the daemon

The daemon is the link between client and server side. It will be started on the server side and observes the predefined exchange directory. As soon as the client side SmartDX process creates a new subdirectory to this directory with the correct naming convention the daemon renames the subdirectory, verifies whether too much SmartDX server processes are running at the same time and, if not, it starts the server side process.

To setup the daemon the parameters can be defined in the config – file located in the same directory executable is stored.

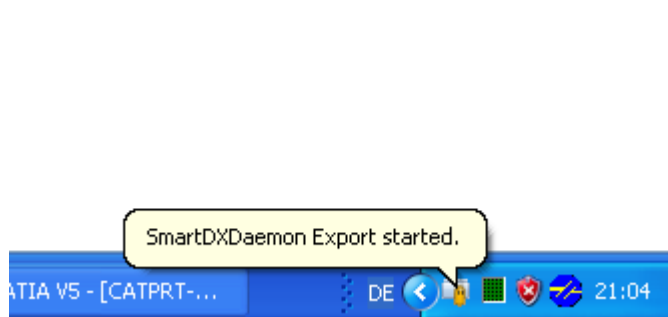
There are two parameters you should modify according to your installation:

- add key="Program" value → The path to the file called runCallFunction.cmd
- add key="Path" value → Path to your defined exchange directory (same as defined in Admin tool)

Each time a daemon indicates a new directory and starts the server side process this action will be reported in log file.

To automate the daemon start during system start you can place the executable and the config file in the startup directory of a dedicated user.

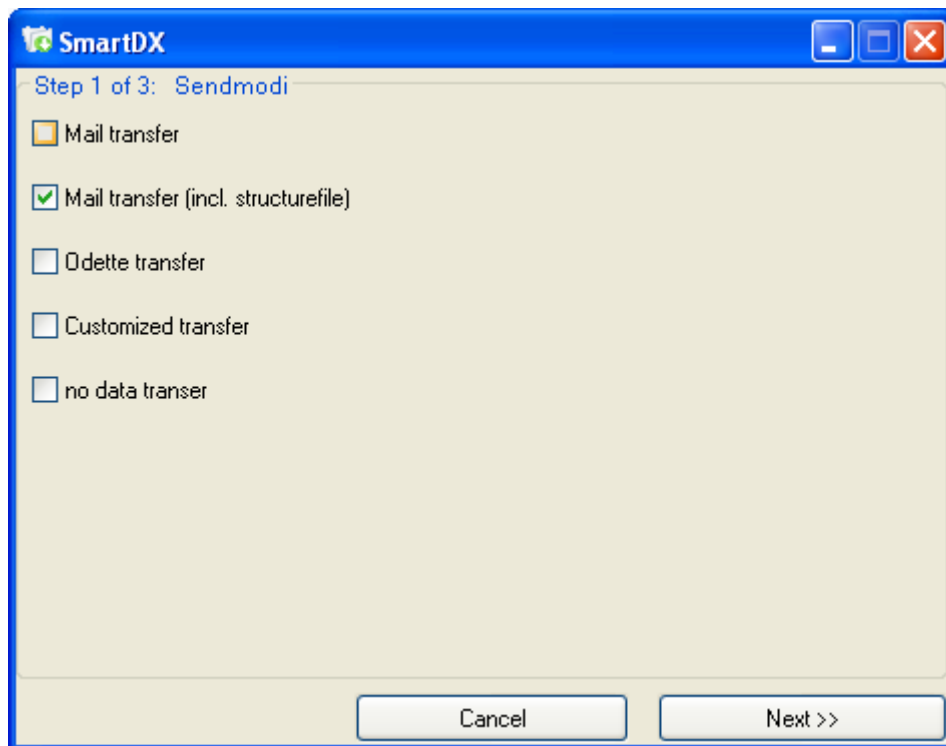
In order to perform the start of the daemon you have to run the 'GUIDaemon.exe'. An icon tray in the lower right corner indicates the daemon is running



3.15 Customizing capabilities with programmable ,Plug Ins'

The server side batch process is a sequential executed application. During this process SmartDX provides several so called Plug Ins to customize the server side behavior. These Plug Ins are libraries delivered in Visual basic 6 source code.

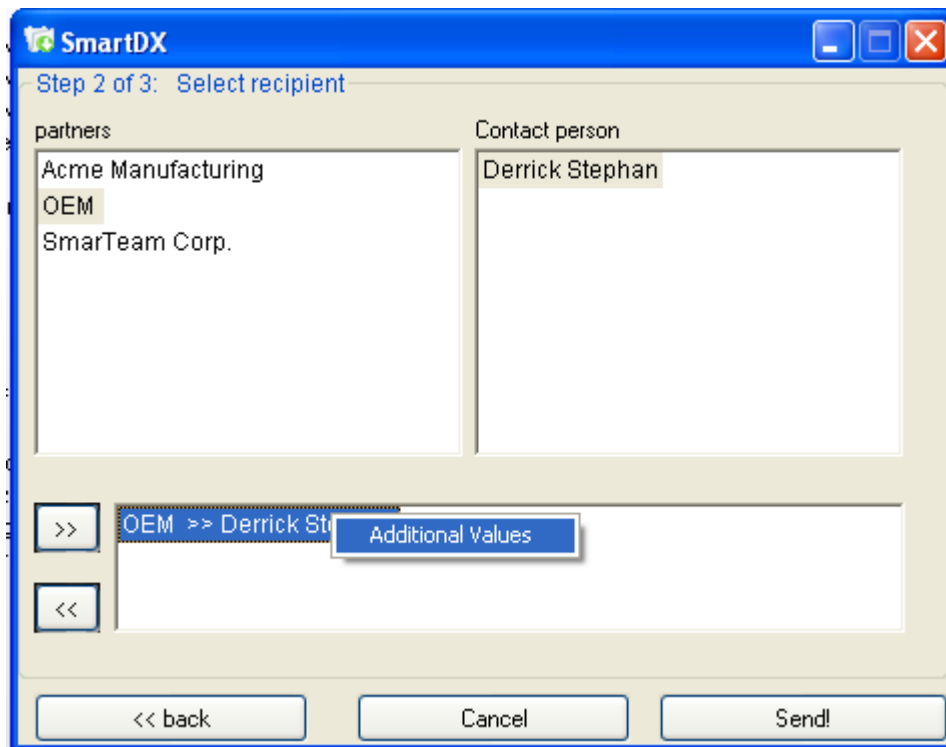
1. CatiaRename_Proj.vbp → Library to create a complex renaming rule to rename the CATIA documents.
2. BeforeCADBatch_Proj.vbp → Library that enables the customization prior to the CATIA batch process.
3. SmartDX_SendCustom.vbp → Library to create a custom specific transfer mode (transfer via ftp, upload to portal, ...). Used if option 4 has been selected in the SmartDX wizard: *customized transfer*
4. SmartDX_OdetteSend.vbp → Library to create a transfer mode using the Odette protocol. Used if option 3 has been selected in the SmartDX wizard: *Odette transfer*.
5. SmartDX_SendMail_Proj.vbp → Library to create an Email transfer mode using the SMTP server. Used if option 1 or 2 has been selected in the SmartDX wizard: *mail transfer*.



6. StructureFile_Proj.vbp → Library to provide the capability to generate a different structure file format than the xml based structure file (i. e. ENGDAT V3 or STEP AP214 cc6).
7. SmartDX_Zipping_Proj.vbp → Library to provide the capability to compress the exchange data using Gzip, tar, winzip, etc. Exemplary a compress method is provided using pkzip. You can use this for testing purpose. However, the executable are not part of the deliverables, but you can download the shareware on the pkzip homepage.
8. FinishDX_Proj.vbp → Library that enables the customization / clean up at the end of the data exchange process
9. SmartDX_ErrorAppl.vbp → Library that enables to react on interruptions during SmartDX server site batch routines. I. e. you can use this Plug In to send an email in case of an SmartDX server interruption.

On the client side you have the capability to use one Plug In for customization. The VB6 project for this Plug In is called 'AddValue_Plugin.vbp' and can be performed during the selection of the recipients. You can use this Plug In for any intercommunication with the user during the SmartDX client wizard. It could be an approach to use this Plug In for additional information / decisions related to the recipient the user has to give. One then can create an additional file containing the results of the Plug In and store it in the data exchange directory. This file can be used to impact the server side batch routine.

The Plug In will be called using the contextual menu of the selected recipient.



Call the 'Additional Values' Plug In

3.16 Customizing the different transfer modes

The administrator is able to define whether a transfer mode will appear to the user or not. Furthermore the administrator can define another caption for the transfer mode (like ,Company A transfer mode')

3.16.1 Hide or show different transfer modes to the user

Under the chapter *AvailableSendModi* the Admin Tool lists the available transfer modes. The value for each transfer mode can be set to 0, 1 or 2.

0. The modus will not appear to the user in the SmartDX wizard.
1. The modus will appear to the user in the SmartDX wizard.
2. The modus will appear to the user in the SmartDX wizard and is the predefined default transfer mode.

3.16.2 Define Error and abnormal termination behavior

Each Plug In provides the capability to define it's own behavior in case of error or abnormal termination. You can define whether to continue or stop the server side process. The settings to declare the behavior are provided with the SmartDX_CallFuncitons.ini under the chapter [StopOnError]. If you assign true to a specific Plug In it will stop the whole server

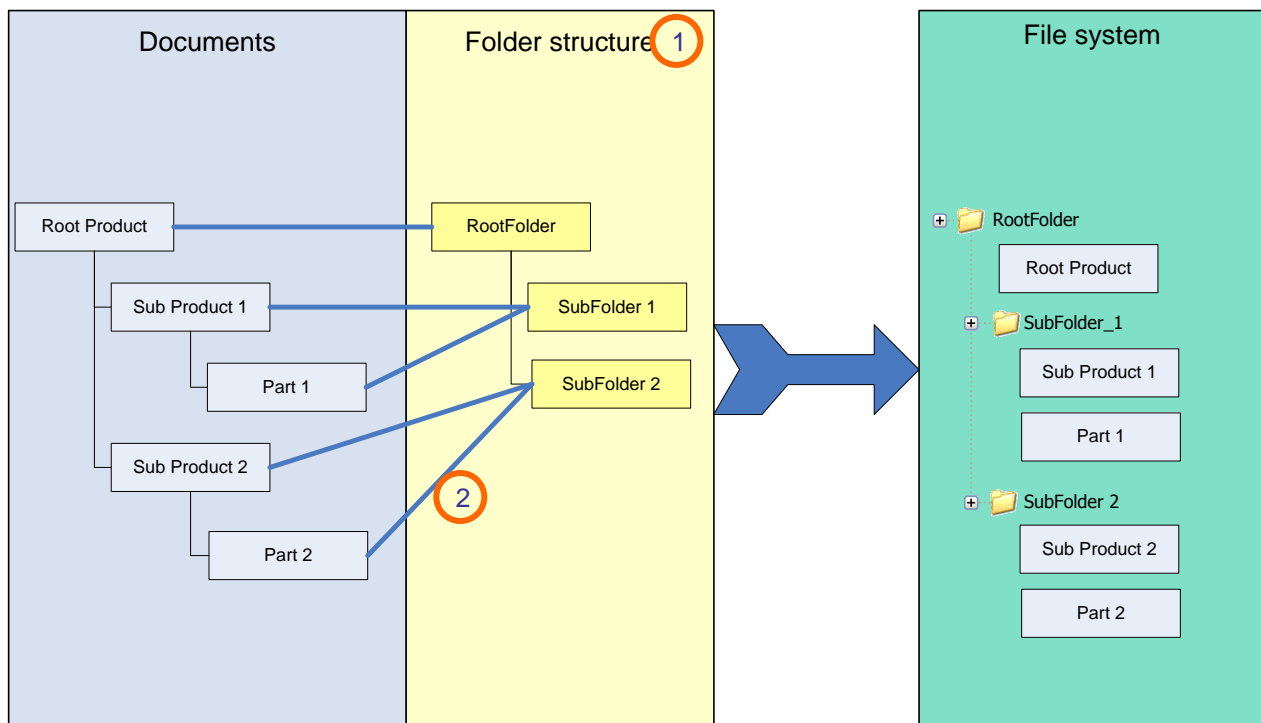
Example:

```
[StopOnError]
CATIABatch=True
CATIARename=True
SturctureFile=True
CompressingFile=True
SendMailFile=True
SendCustomFile=True
SendOdetteFile=True
```

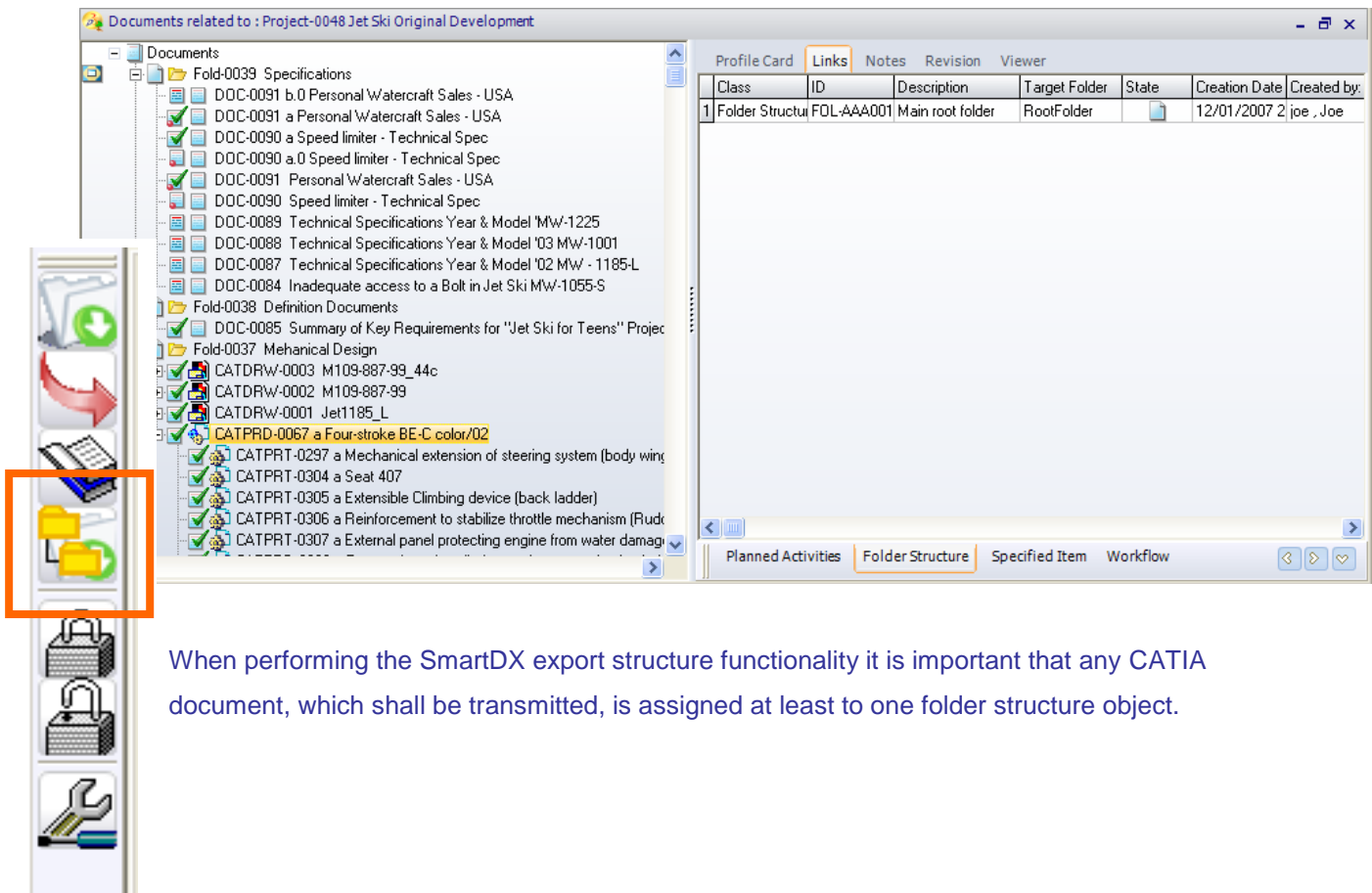
4 SmartDX enhancements

4.1 Customizing the SmartDX Export structure functionality

The purpose of the SmartDX Export structure functionality is to provide a capability that stores the CATIA documents not in a flat structure but in a predefined folder structure on the file system

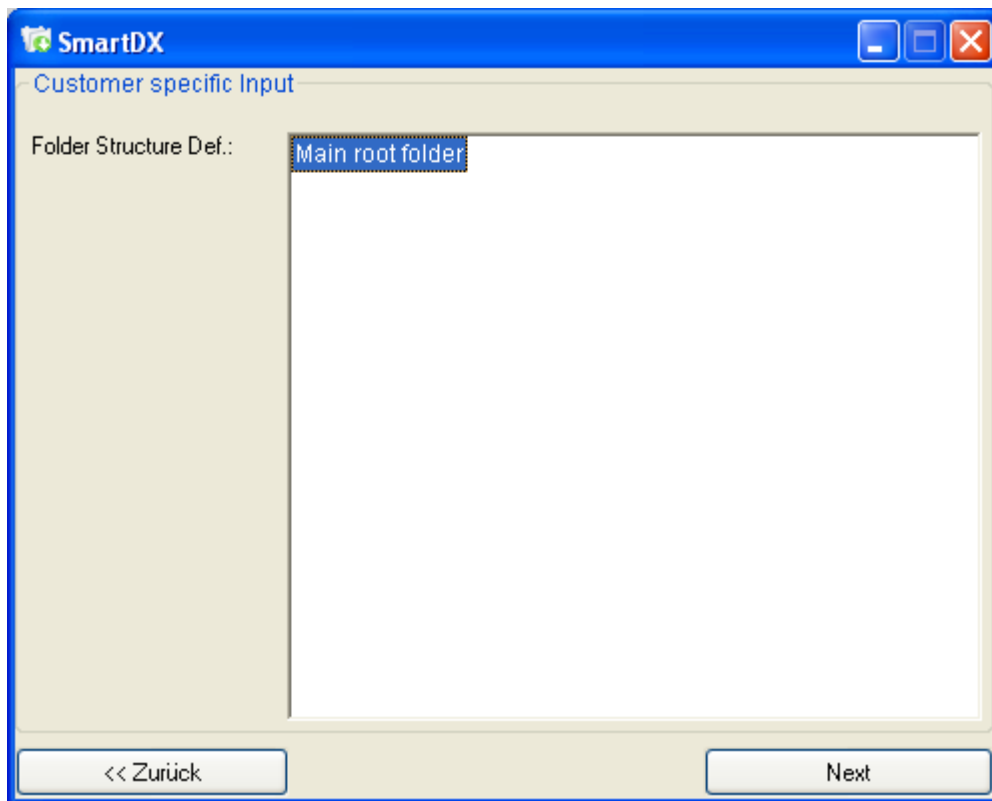


Before running the SmartDX export structure you have to predefine a folder structure in SmarTeam (1) using the object class you assigned the class mechanism 'folder structure' to. This folder structure represents the folder structure on the file system side. In the next step you are going to assign each document you want to store related to a folder on the file system side to the folder you defined in ENOVIA SmarTeam® using a general link (2).



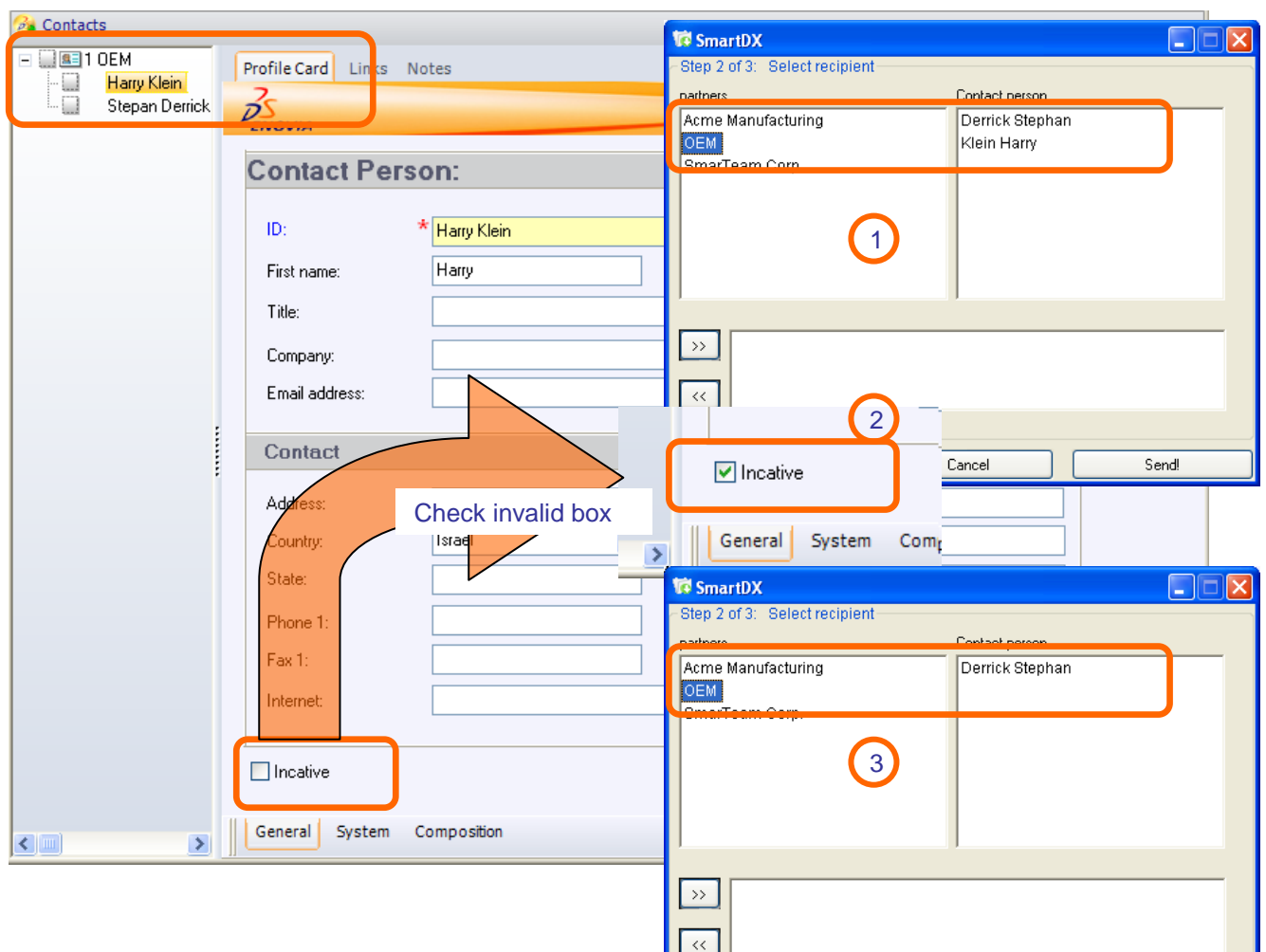
When performing the SmartDX export structure functionality it is important that any CATIA document, which shall be transmitted, is assigned at least to one folder structure object.

During the SmartDX wizard after selecting the recipient you will be prompted to choose a folder structure definition. This is because you can define several folder structure definitions to one CATIA product structure. It is very important to understand that each child object of the selected folder structure can only be linked to exact one parent. It is not allowed to have multiple parent references.



4.2 Exclude customer / contact person from SmartDX wizard list

In principal all customers and contact persons, which has been generated using the appropriate object class, will be listed in the SmartDX wizard. To exclude customers or contact persons from this list, it is not necessary to delete the SmarTeam object. Therefore an attribute (TDMX_DX_INACTIVE) can be used. In case you want a customer or a person to be excluded from the list in the wizard just check this attribute and the appropriate customer / contact person will not appear to the user.



4.3 Customer specific form in SmartDX wizard

To request additional, custom specific information from the user, which can be used on the server side batch process later on, the SmartDX wizard provides an additional, customizable form.

This custom form can only be used in combination with the transfer mode *Custom Send mode*.

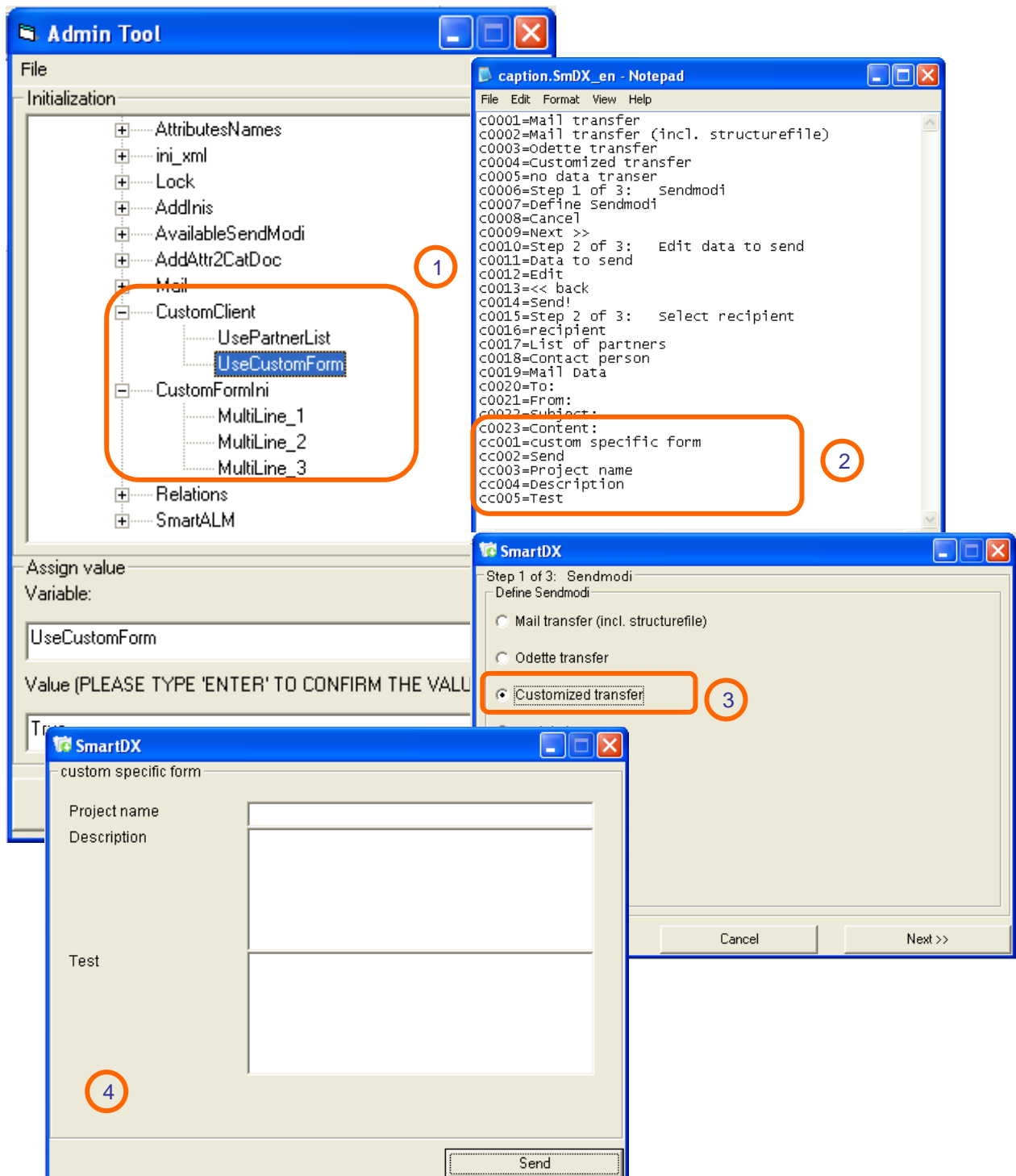
To use the capability the administrator has to set the parameter *UseCustomForm* in the chapter *CustomClient* to true. Next the user chose the *custom send mode* the custom form will appear to him.

In the chapter *CustomFormIni* the amount of input boxes can be defined. Depending on your needs you can either define multi line boxes or single line boxes. There is no limitation to the amount of boxes except for the window space. Each text box that should appear has to be defined in the chapter *CustomFormIni* by a parameter called *MuliLine_x* (where x means the continuously number of lines). The value for each textbox parameter defines either it is a multi line textbox (true) or a single line textbox (false). The amount or parameters in the chapter corresponds to the line appearing in the custom form.

The caption for each textbox can be defined in the language files (please refer the chapter language files for more information). The custom form related translations in the language files will have the prefix 'cc' instead of 'c'. The first parameter cc001 is reserved for the form caption and the second one for the button caption.

| | | |
|-----------------------------|--|--|
| cc001=Custom specific input | => Form caption | |
| cc002=Send | => Button caption | |
| cc003=Project name | } Captions for input textboxes cc003 – c00x | |
| cc004=Description | | |
| cc005=Test | | |
| cc00x=... | | |

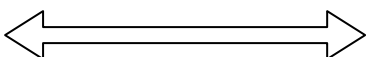
The input values the user will do during the SmartDX wizard will be enclosed in the SmartDX_Data.xml that is stored in the exchange directory. You are able to call them using the Plug Ins. The values will be assigned by the parameters CustomText_1 to CustomTeyxt_x.

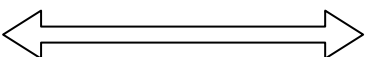


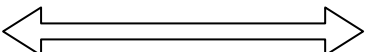
4.4 Disable link between data ex. Obj. and transmittal or custom obj.

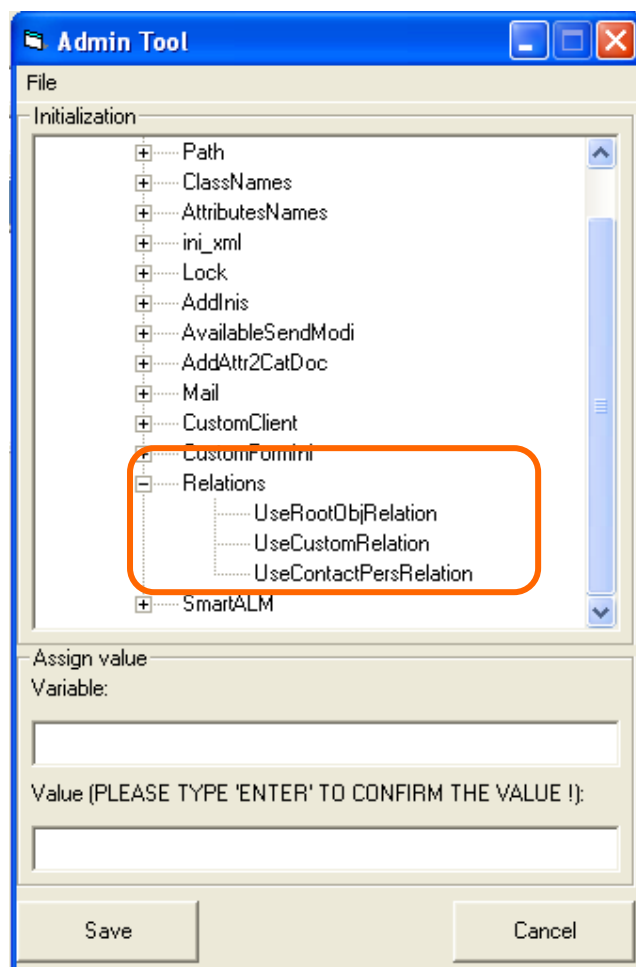
The link generation between the data exchange object and the root transmittal or custom / contact person object is by default enabled. However, you have the capability to disable the generation of those links using the Admin Tool.

In chapter *Relations* you can define the link behavior for each link capability

Data exchange object  Root transmittal object

Data exchange object  Customer object

Data exchange object  Contact person object

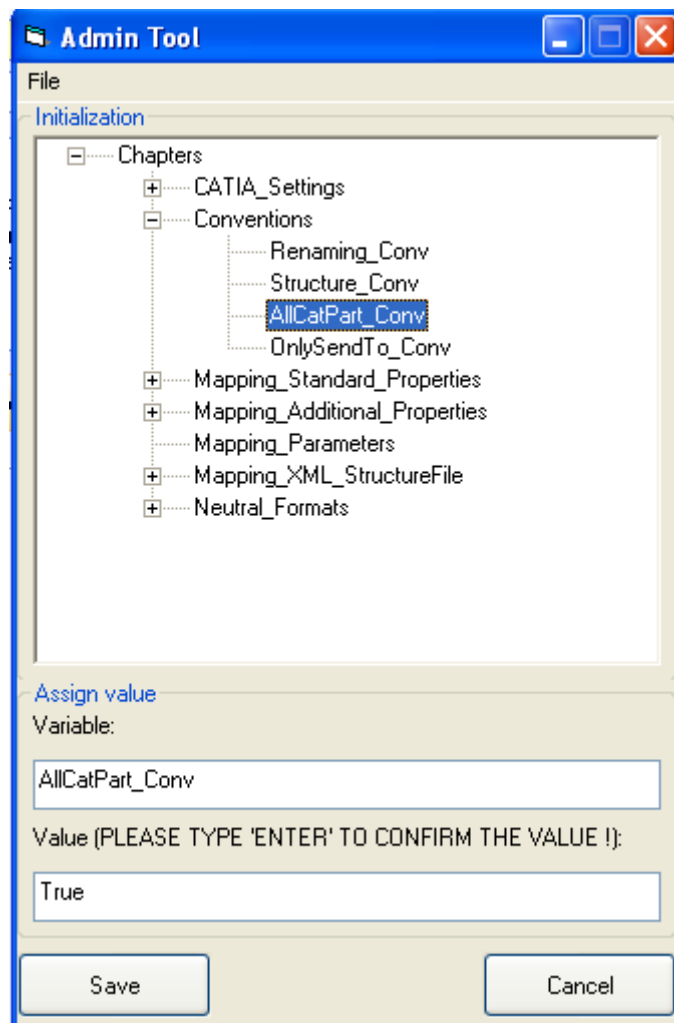


5 What's new

5.1 DROP 6: CATIA Product to Part (All CAT Part) generation

Since Drop 6 the capability is provided to generate a so called AllCATPart based on the selected CATIA root documents. This enables the sender on the one hand to compress a CATIA Product into a single CATIA Part, on the other hand this function suppress all development intelligence that has been embedded into the product. For that reason AllCATParts are often used in the automotive branch.

To enable the use of AllCATPart generation you only have to set the parameter AllCatPart_Conv for the appropriate recipient to 'True'.



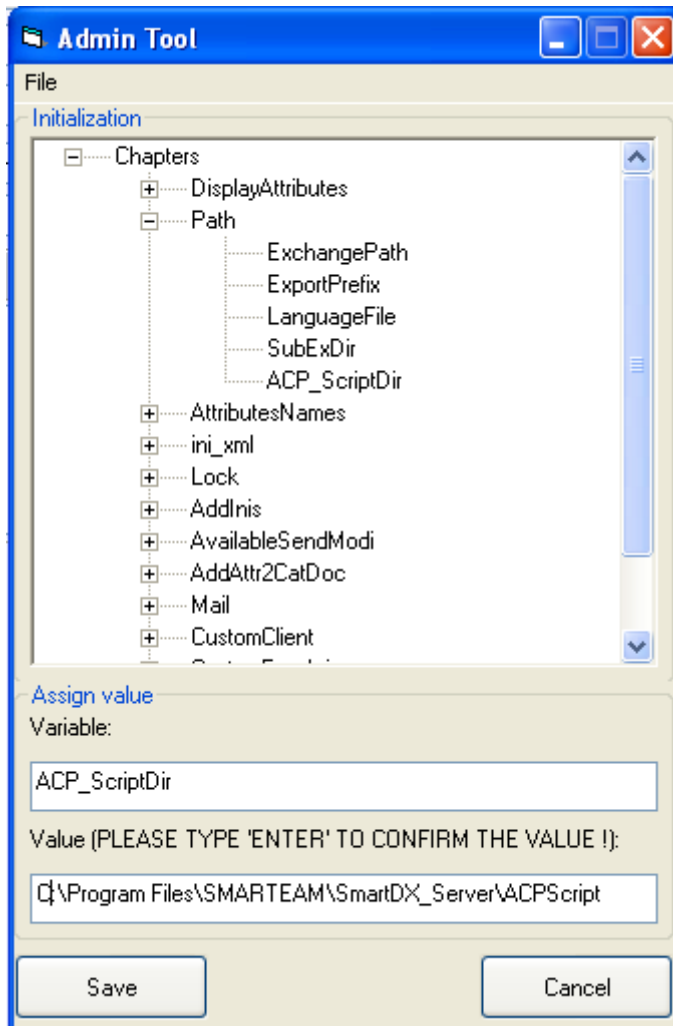
In the Admin Tool you also have to set the right path to the SmartDX_ACP.catvba Script that is provided during SmartDX Server installation (SX9). Normally this directory should be this one:

SmartDX - Implementation Guide – V5.8

Do not reproduce, copy or use without a license from DASSAULT SYSTEMS

© 2010 DASSAULT SYSTEMS Deutschland, All Rights Reserved.

Page 75 / 85

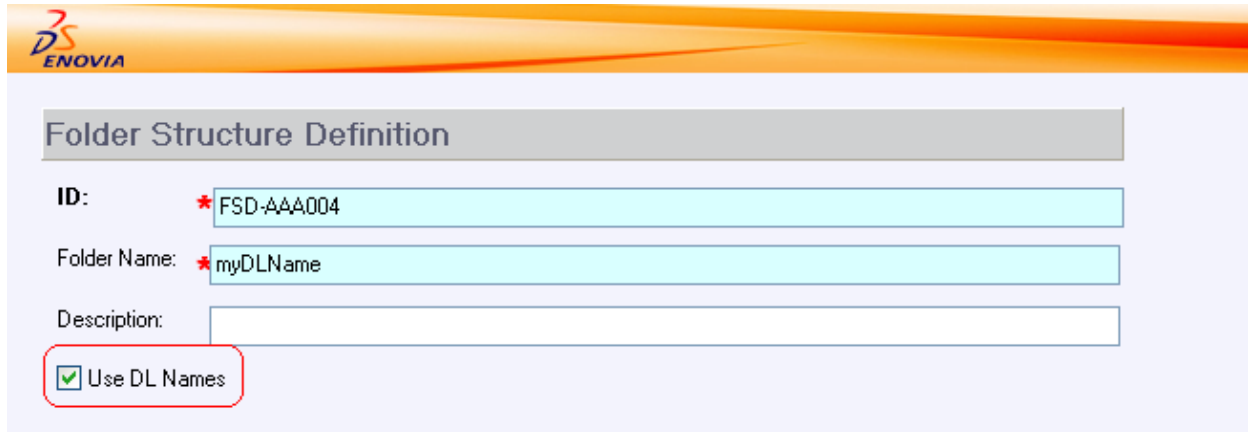


In the export directory you will find in the exchange sub directory beside the CATIA root Product and its related CATIA documents a CATIA single Part that is named likely the root Product with the name *RootProductName_AllCATPart.CATPart*.

5.2 DROP 6: Exporting CATIA documents related to DL names

It is often required by the recipient to transmit CATIA documents that are related to a predefined so called DL name folder. SmartDX provides the capability to save all CATIA documents per data export process in a predefined, assigned DL name folder using CATIA's 'Send to' functionality. The document will be stored underneath the DL name folder at is related to it. The definition is similar to the folder structure definition.

Each CATIA document that should be stored in a DL name folder during export process must be linked to a folder structure object that contains the DL name folder name.



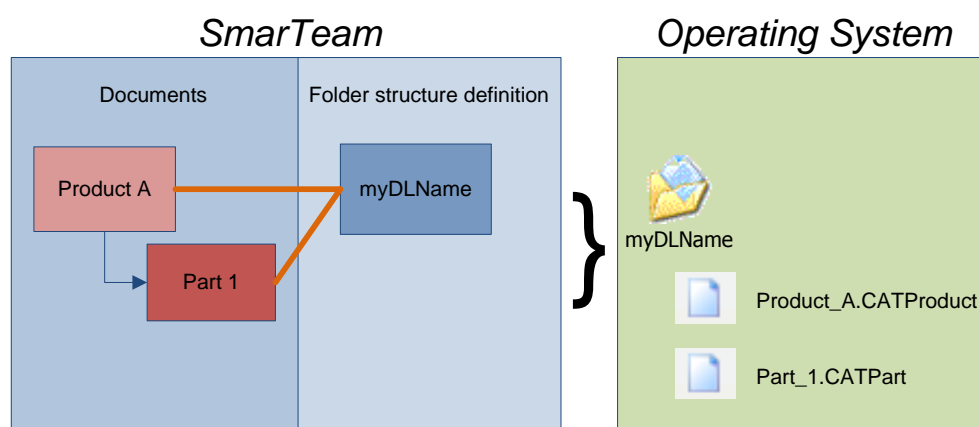
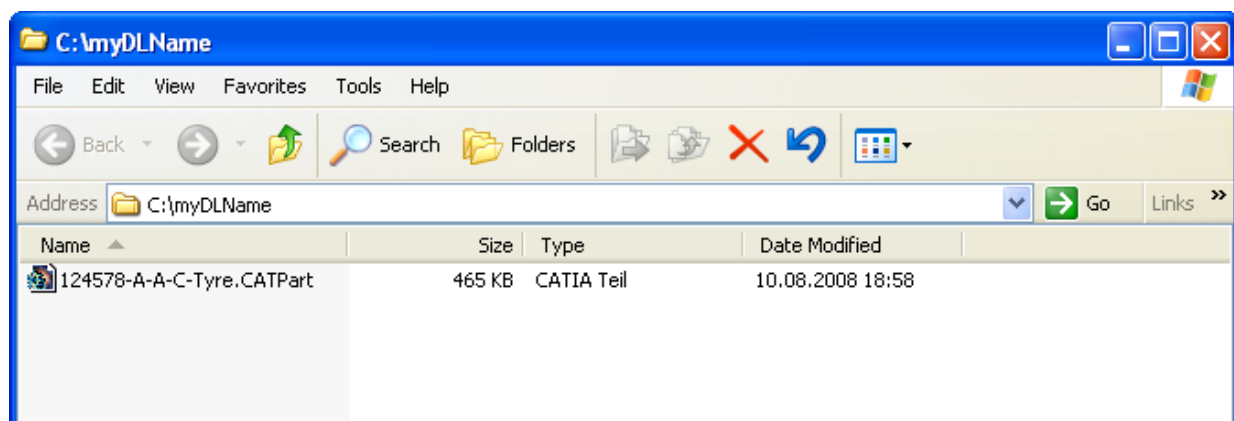
Folder Structure Definition

ID: * FSD-AAA004

Folder Name: * myDLName

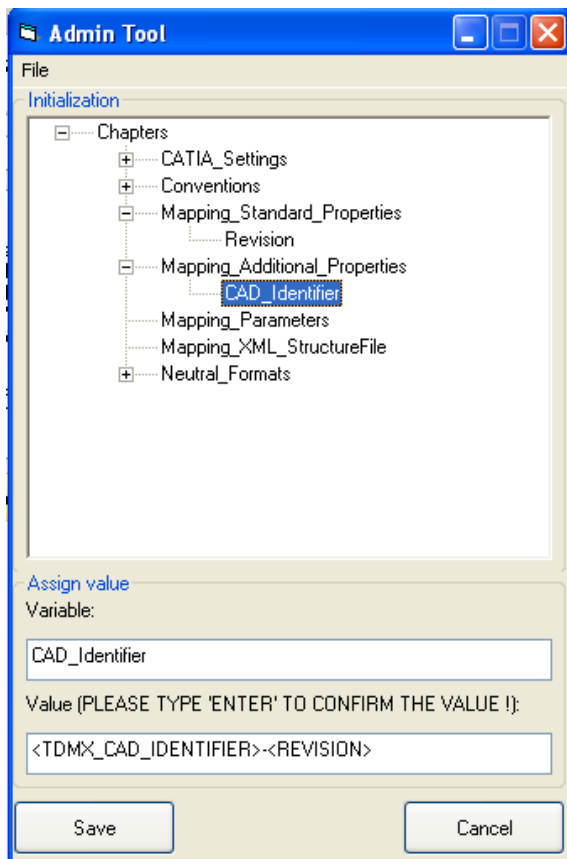
Description:

☒ Use DL Names



5.3 DROP 6: Multi attribute mapping for CATIA properties / parameter

You are able to assign more than one SmarTeam attribute in the SmartDX – CATIA settings. Each attribute definition will be specified in sharp brackets. Characters defined between those brackets are used as separators



5.4 DROP 7: Observe the server site process on client site

The user often asks to have a quick overview of the progress and the current status of the SmartDX server site process in an easy way without looking periodically to the data exchange object in SmarTeam. Therefore with Drop 7.0 a new functionality is available.

After the user finishes with the SmartDX wizard an icon will appear in the tray icon bar (normally in the lower right corner of the screen). Different colors of this icon indicate the different states of the SmartDX server site process. The help bubble provides more detailed information with respect to the server site process. By right click on the icon and selecting 'Show' menu the user will see a list of the process history.



To enable this functionality the parameter 'UseServerMonitor' in chapter 'Settings' in the Admin tool has to be set to 'true'.

Ensure, that the entire icon and the executable that are needed are located in the path where the SmartDX.dll is located.



The light blue icon indicates that SmartDX client wizard has been finished and the job is now to be started on the serve site.

The dark blue icon indicates that the process is on work. To get more infos about the current batch process just put the mouse arrow on top of the icon and the help bubble will appear to display this information.

If you have enabled the break capability for validation purposes for example then when achieving this state the color turns into orange.

After completely finishing the server site process the color went into green with a white check mark inside.

In case an error occurs the icon turns into red with a white cross inside.

5.5 DROP 7: New 'Plug In' to react on error interruptions

In case of an error during the SmartDX server site process an error event will be raised and this will call the new 'Plug In' to enable customization to react on this situation. For example administrators can be informed by mail. This new 'Plug In' can be found in \SMARTEAM\SmartDX_Server\Plug Ins\ExtErrorAppl as VB6 project and can be customized to suit the customers' demands.

5.6 DROP 7: ‘ Plug In’ BeforeCATIABatch has been enhanced to build custom specific CATIA start strings

The optional parameter ‘CATIAStartString’ is an output parameter and returns a CATIA Start string that will be used to start CATIA during the following CATIA batch routine. If you want to use this new parameter to over give a customized start string please ensure that it will include the parameter for CATIA batch mode starting with a macro. It should look like the following:

```
"C:\Program Files\Dassault Systemes\B18\intel_a\code\bin\CNEXT.exe" -env  
"CATIA.V5R20.B20" -direnv "C:\Documents and Settings\All Users\Application  
Data\DassaultSystemes\CATEnv" -batch -macro  
"C:\Exchange\Export_joe_20090511_165209_InWork\Temp\Macro.CATScript"
```

The parameter ‘ExchangePath’ in BeforeCATIABatch ‘Plug In’ includes the value for the appropriate exchange folder.

5.7 DROP 7: CATIA batch process interruption after predefined idle time

As CATIA is not fully batch mode enabled it could happen in seldom cases that CATIA is waiting for user input for example. To avoid that such process will ‘hang’ until they will be stopped manually a new capability is now available to shut down the CATIA batch process automatically after a predefined idle time. This idle time will be measured by validating the memory allocation. If the allocated memory for the CATIA process is still the same for a predefined timeframe (by default for 300 seconds) then the CATIA process will be shut down and the error event will raise. To disable this behavior you can set the Idle time to 0 second.

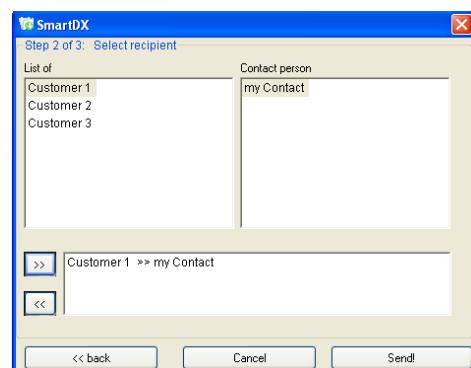
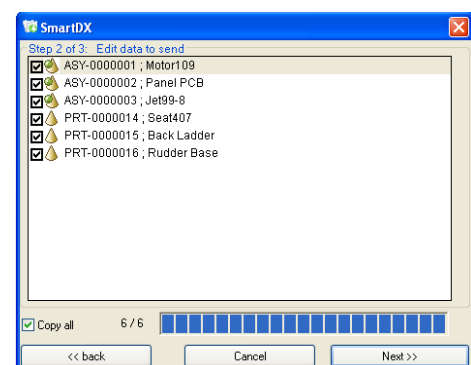
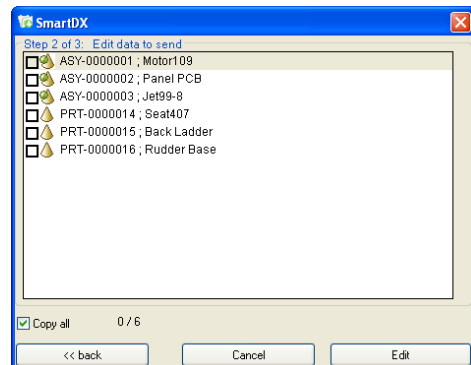
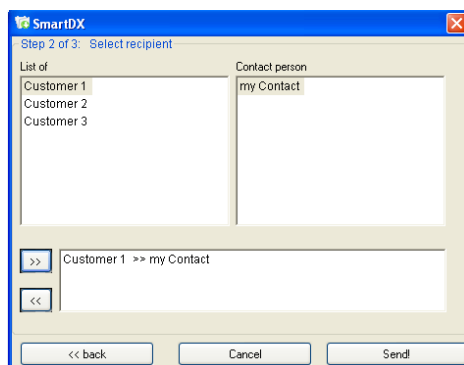
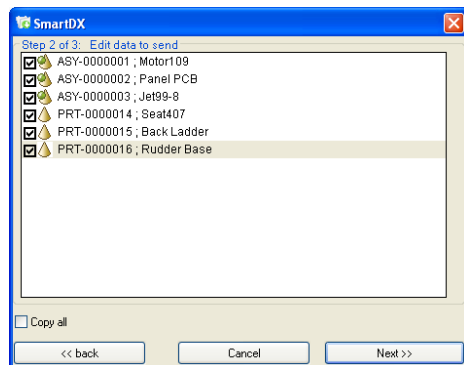
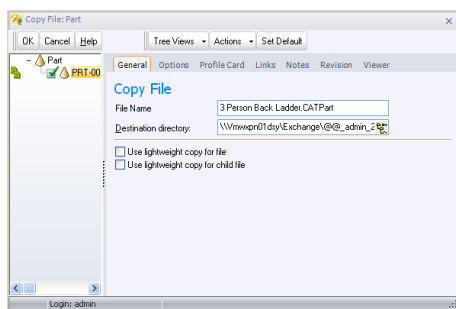
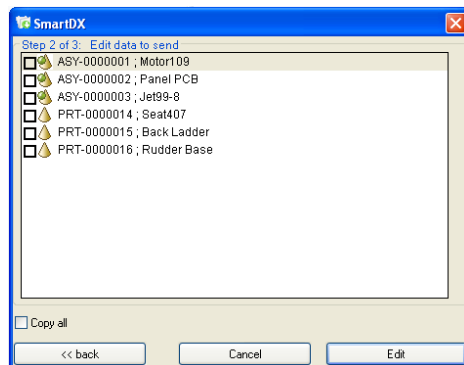
5.8 DROP 7: Enable users to copy all selected files in one step

Normally all selected files will be copied one by one using the default SmarTeam CopyFile operation during SmartDX client wizard. The new function allows the user to copy all selected files in one step without opening the SmarTeam copy file screen. This function will use the default SmarTeam life cycle rules for copy file operation.

To use the function you have to select the check box ‘Copy all’ on the ‘Edit data to send’ screen. A progress bar represents the current status of the copy file operation. After the operation finished all selected objects are checked.

If you want the ‘Copy all’ check box to be preselect by default you can set the parameter ‘PresetCopyAllBox’ to true.

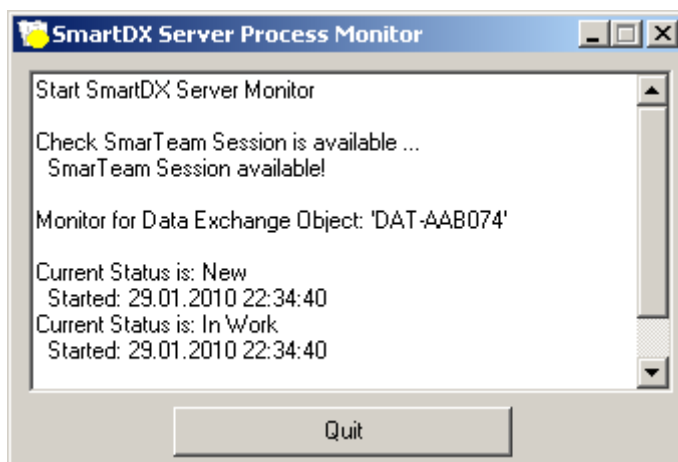
Implementation Guide SmartDX



5.9 DROP 8: Bugfixing and enhancements

5.9.1 SmartDX Monitor enhancement

For easy identification the Report screen of the SmartDX Monitor contains the ID of the related Data Exchange object in SmarTeam



By double click on the SmartDX Monitor Icon in the tray bar the report screen will appear.

5.9.2 Using same string multiple times for renaming CATIA documents

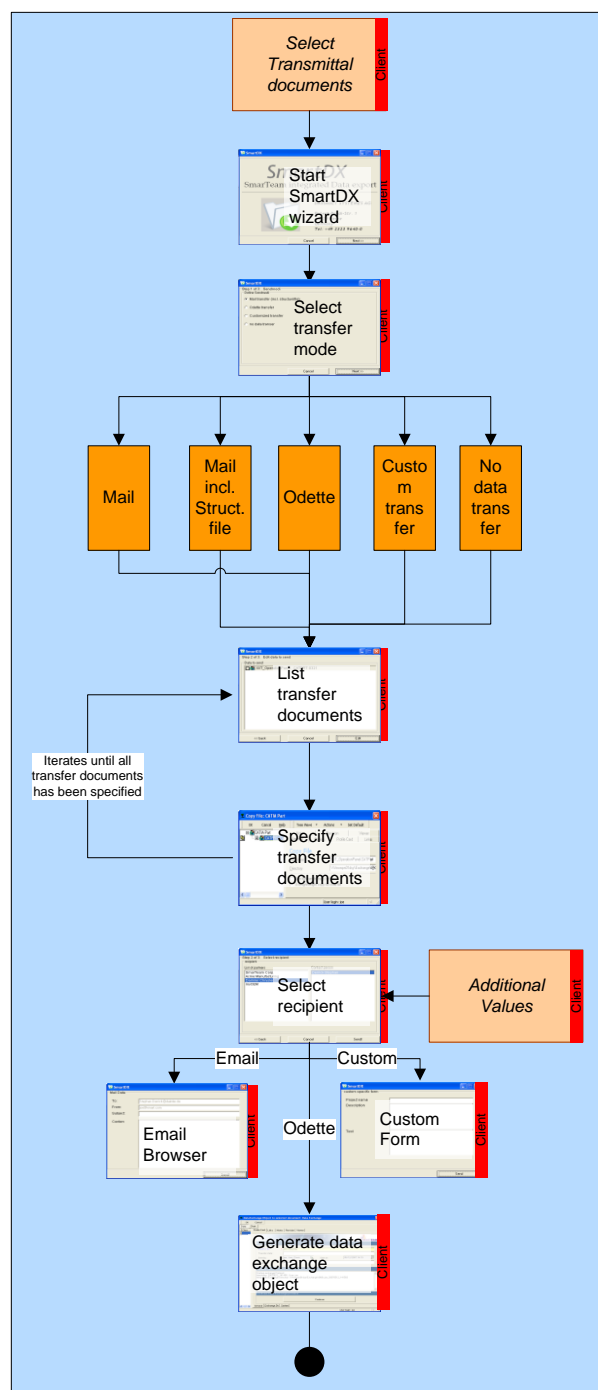
SmartDX server site process now takes care about using the same renaming string multiple times during the CATIA batch process while renaming CATIA documents. After identifying this issue the process will stop with error and the affected CATIA documents will be listed in the report box of the data exchange object.

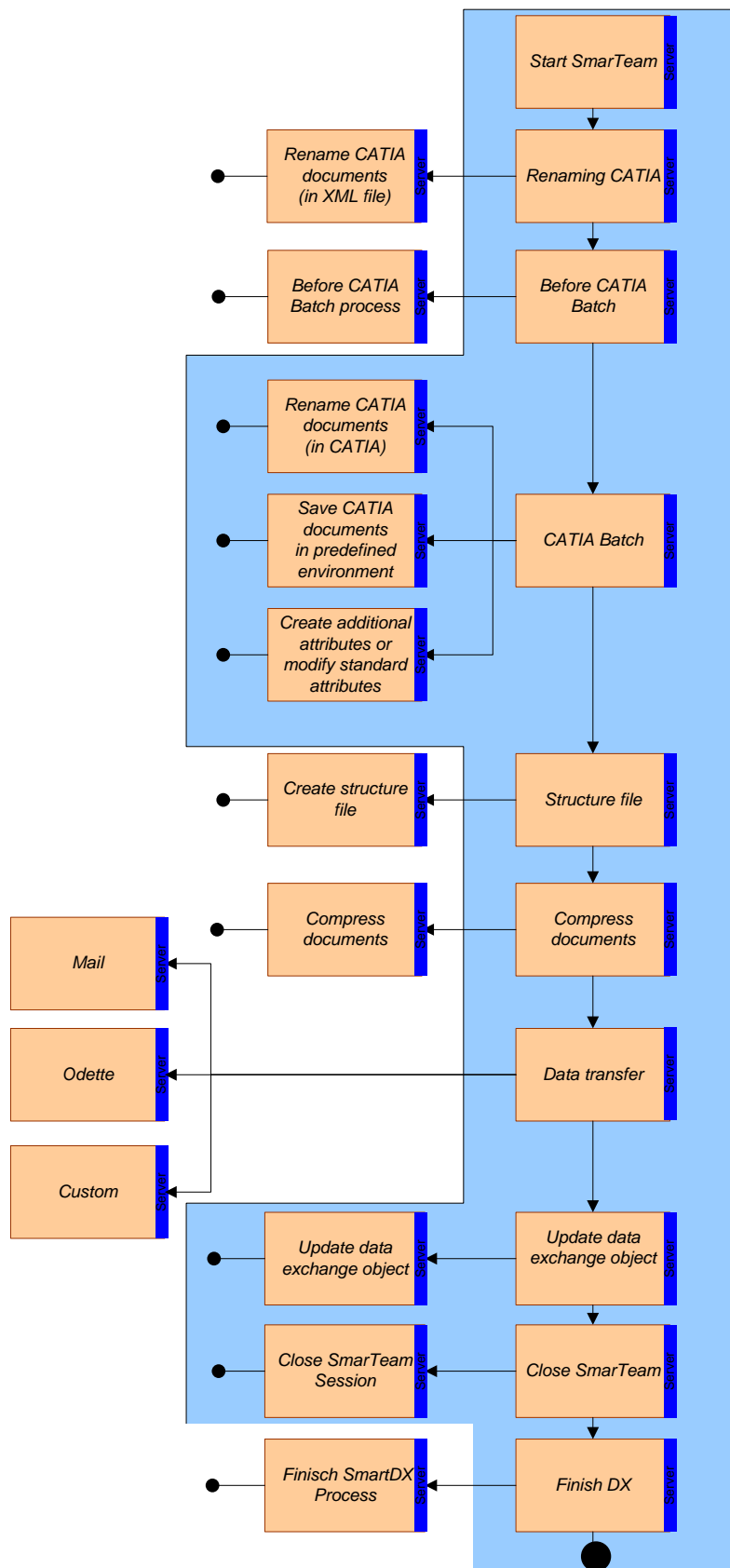
5.9.3 Disable the use of 'copy all' Check box after using the 'Edit' – Button

After performing the Edit – Button in 'Step 2 of 3' – screen of SmartDX client the option 'Copy all' check box will be disabled to avoid accidentally use.

6 SmartDX flowchart

This is to clarify the SmartDX work flow in principle. Each operation is represented either by a red marked box (client side) or a blue marked box (server side).





7 Known Issues

7.1 Defining formats for neutral conversion

In order to use the neutral conversion one has to define the formats in customer related admin object. **This definition has to be finished by a semicolon!** The definition could look like this:

```
stp;igs;model;
```

Otherwise the last format would be ignored

7.2 Using 'AllCatPart' conversion

In case the 'AllCatPart' conversion is enabled it is important **not to interact with the SmartDX server during the batch process**. Otherwise the batch routine will be interrupted!

As the 'AllCatPart' – functionality cannot be guaranteed to run proper for all environments this functionality will be delivered as it is without support and warranty. You can use it on your own risk.

7.3 Using 'AllCatPart' in R19

There are known issues for using the 'AllCatPart' – functionality in combination with CATIA R19 GA and CATIA R19 SP01.

7.4 CATIA batch process interrupts when CATIA Parts including Sheet Metal Design features

This happens when CATIA batch process will launch CATIA Parts that includes Sheet Metal Parts without having this product installed and / or no valid license for Sheet Metal Design. The CATIA batch process is waiting for user interaction to confirm to proceed the process. To avoid this behavior you can set the following environment variable:

```
DEACTIVATE_SHEETMETAL_WARNING_AT_PART_OPEN=YES
```