



ENOVIA SmarTeam

SmarTeam – MatrixOne Integration Implementation Guide

© Dassault Systèmes, 2008. All rights reserved.

CATIA, ENOVIA, SMARTEAM and the 3DS logo are registered trademarks of Dassault Systèmes or its subsidiaries in the US and/or other countries.

PROPRIETARY RIGHTS NOTICE: This documentation is the property of Dassault Systèmes. This documentation shall be treated as confidential information and may only be used by employees or contractors of the Customer in accordance with the terms of the End-User License Agreement accepted by Customer.

Any use of the Licensed Program contained in this media or accompanying it, is subject to the terms of the End User License Agreement accepted by Customer. The Licensed Program is protected by international copyright laws and international treaties. Unauthorized use, reproduction and/or distribution of any of the Licensed Program, or any part thereof, may result in severe civil and/or criminal penalties, and will be prosecuted to the maximum extent possible under the law. Company names and product names mentioned herein are the property of their respective owners and certain portions of the Licensed Program contain elements subject to copyright owned by these entities. See the Documentation CD provided with the Licensed Program for details and/or additional terms and conditions relating to these entities.

Part No: GEN-I4-190208

Table of Contents

Chapter 1: Overview	1
What is SmarTeam – MatrixOne Integration?	1
Architecture	2
Version Compatibility	2
Related Documentation	2
Internet Site	3
Chapter 2: SmarTeam – MatrixOne Integration Checklist	4
Chapter 3: Installing SmarTeam – MatrixOne Integration	5
Prerequisites	5
Running the Installation	5
Chapter 4: Configuring SmarTeam – MatrixOne Integration	8
Connecting to SmarTeam	8
Configuring Users	8
Mapping Users	9
Configuring UserMappingFilePath	9
Mapping Classes and Attributes	9
Class Mapping	10
Attribute Mapping	10
Configuring schemaMappingFilePathLocation	10
Chapter 5: Configuring MatrixOne	12
System Software Requirements	12
Configuring the params.xml File	12
Configuring MatrixOne	13
Chapter 6: Working with SmarTeam – MatrixOne Integration	15
Working with SmarTeam – MatrixOne Integration	15
Limitations	15
Chapter 7: Troubleshooting	16
Known Issues	16
Frequently Asked Questions	16

Chapter 1: Overview

This guide provides all the information necessary for a system administrator to work with the an ENOVIA SmarTeam product and an ENOVIA MatrixOne product as one unified PLM system by using SmarTeam – MatrixOne Integration.

What is SmarTeam – MatrixOne Integration?

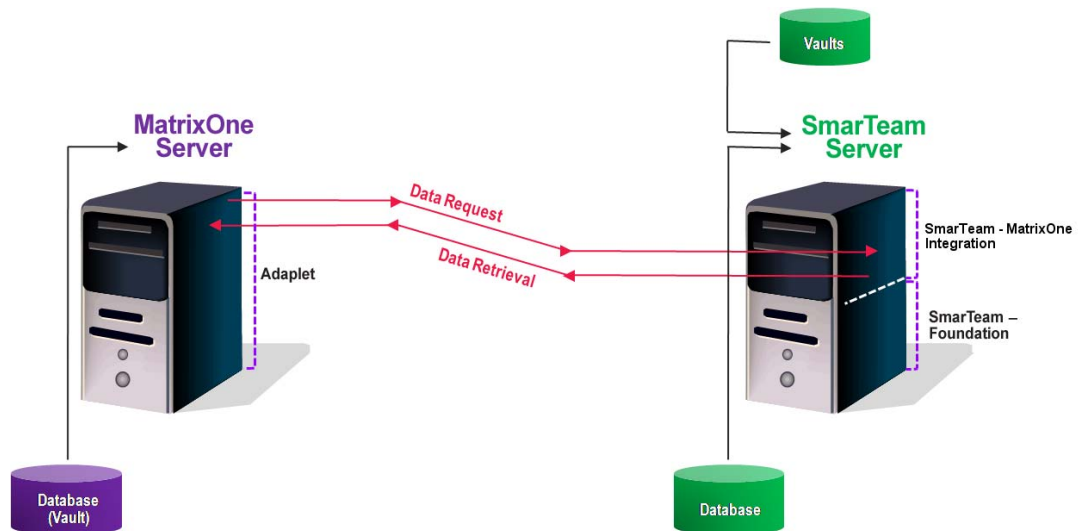
SmarTeam – MatrixOne Integration is a connector, which is hosted on a SmarTeam server along with SmarTeam – Foundation and enables a MatrixOne user to view SmarTeam data via MatrixOne. The origin of the SmarTeam data is transparent to the user. Data can be retrieved by the MatrixOne user from SmarTeam, but not sent to SmarTeam.

MatrixOne and the SmarTeam – MatrixOne Integration must be configured and mapped correctly to enable MatrixOne users to view SmarTeam data. The following must occur:

- SmarTeam – MatrixOne Integration must be installed on the SmarTeam server (see [Installing SmarTeam – MatrixOne Integration](#))
- MatrixOne must be configured so data from SmarTeam can be transferred to MatrixOne (see [Configuring MatrixOne](#))
- SmarTeam – MatrixOne Integration must be configured to connect to SmarTeam and can enable mapping between SmarTeam users and classes to MatrixOne users and classes (see [Configuring SmarTeam – MatrixOne Integration](#))
- After SmarTeam – MatrixOne Integration is installed and configured, and MatrixOne is configured, the MatrixOne user is ready to view and work with SmarTeam data from the MatrixOne interface via SmarTeam – MatrixOne Integration (see [Working with SmarTeam – MatrixOne Integration](#))

Architecture

The following image represents the SmarTeam – MatrixOne Integration architecture.



- SmarTeam server hosts SmarTeam – Foundation and SmarTeam – MatrixOne Integration. Data is retrieved from the SmarTeam database via the SmarTeam API / server. Files are sent from the SmarTeam Vaults to the SmarTeam server.
- SmarTeam – MatrixOne Integration is a connector, which sits on the SmarTeam server, and connects the MatrixOne adaplet with the SmarTeam server to enable the transfer of data to MatrixOne. (Data flows in only one direction, from SmarTeam to MatrixOne.) The SmarTeam – MatrixOne Integration receives data requests from the MatrixOne adaplet, retrieves data and files from the SmarTeam database and SmarTeam Vaults via the SmarTeam server, and sends the data to MatrixOne.
- MatrixOne server includes the MatrixOne adaplet, which is a feature that enables MatrixOne to communicate with Web Services similar to the SmarTeam – MatrixOne Integration.
- MatrixOne stores the metadata in vaults and files in FCSs.

Version Compatibility

SmarTeam – MatrixOne Integration version R19 SP2 works with MatrixOne version V6R2009.

Related Documentation

The following documents are referred to in this guide and are available on the SmarTeam Documentation CD.

Document	Remarks
SmarTeam – Foundation Installation Guide	Provides installation procedures to customize and maintain SmarTeam – Foundation.
SmarTeam – Web Editor Installation Guide	Provides installation procedures to customize and maintain SmarTeam – Web Editor. In addition, provides information regarding the System Configuration Editor
SmarTeam – Editor Installation Guide	Provides installation procedures for IIS.

Internet Site

You are highly recommended to frequently visit our website for the latest updates and plug-in products, including the latest Service Packs, Program Directory (Release Notes), Hotfixes and Technical Support at <http://support.smarteam.com/>.

In addition, you will also be able to view any installation known issues.

Chapter 2: SmarTeam – MatrixOne Integration Checklist

This checklist provides a detailed list of all the steps that need to be performed and the order in which the products need to be installed. You must complete all the stages in this checklist to successfully install SmarTeam – MatrixOne Integration software.

*Requirement: M = Mandatory, O = Optional

	Item	M/O*	Reference
Stage 1: Prerequisites			
<input type="checkbox"/>	Verify that your software meet the requirements of all other products listed as a prerequisite	M	SmarTeam Hardware and Software Requirements Guide
<input type="checkbox"/>	Install SmarTeam - Foundation version R19 SP2	M	SmarTeam - Foundation Installation Guide
<input type="checkbox"/>	Install IIS	M	SmarTeam - Editor Installation Guide
<input type="checkbox"/>	Install the System Configuration Editor for configuring the system configuration keys	M	SmarTeam - Web Editor Installation Guide
<input type="checkbox"/>	Check for any additional prerequisites on the SmarTeam Web Site	M	Release Notes of Latest Service Pack in the Release or SmarTeam Support Web Site
Stage 2: Installation Process			
<input type="checkbox"/>	Install SmarTeam – MatrixOne Integration on SmarTeam – Foundation	M	Running the Installation
Stage 3: Configuration			
<input type="checkbox"/>	Configure SmarTeam – MatrixOne Integration to connect to SmarTeam and to enable mapping between SmarTeam users and classes to MatrixOne users and classes	M	Configuring SmarTeam – MatrixOne Integration
<input type="checkbox"/>	Configure MatrixOne so data from SmarTeam can be transferred to MatrixOne	M	Configuring MatrixOne
Stage 4: Work with SmarTeam – MatrixOne Integration			
<input type="checkbox"/>	Enable viewing of SmarTeam data	O	Working with SmarTeam – MatrixOne Integration

Chapter 3: Installing SmarTeam – MatrixOne Integration

This chapter provides a step-by-step description of the installation process for installing SmarTeam – MatrixOne Integration.

Buttons

These buttons are available at the bottom of the installation windows:

- **Next:** Confirms your selection in the current window and proceeds to the next window
- **Back:** Enables you to return to the previous window
- **Cancel:** Enables you to abort the installation setup program without installing the SmarTeam – MatrixOne Integration

Prerequisites

Before installing SmarTeam – MatrixOne Integration, you must perform the following steps in the order specified:

- Install SmarTeam – Foundation, version R19 SP2
- Install IIS
- Install the System Configuration Editor for configuring the system configuration keys

Running the Installation

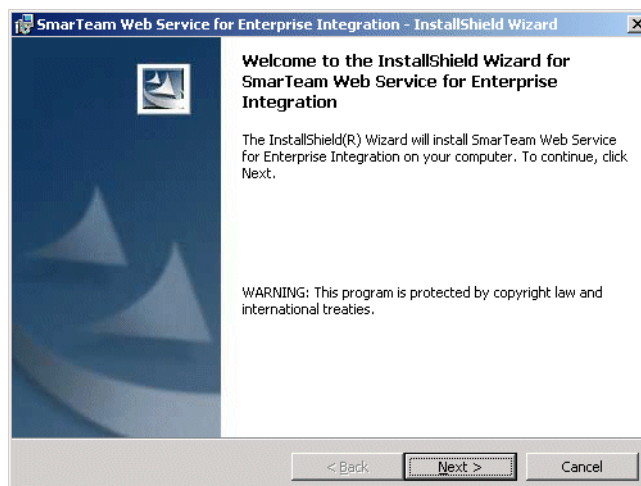
To begin installing:

- 1 From a machine that also has SmarTeam – Foundation installed, run **Setup.exe** from the SmarTeam – MatrixOne Integration software.

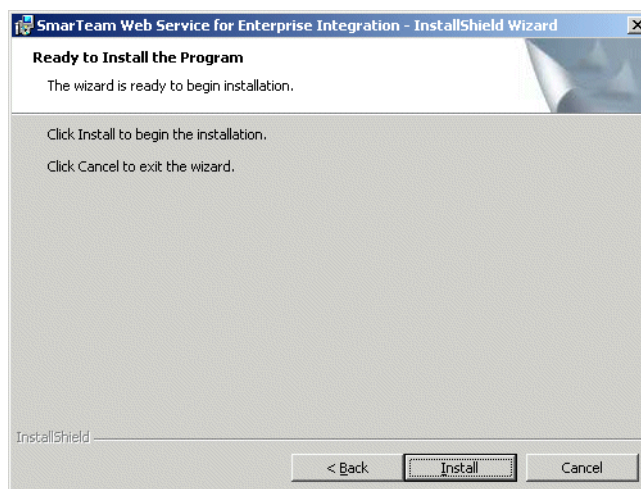
The installation application is launched automatically and appears on your screen to take you through the software installation process.

- If for some reason the installation program does not appear on your screen, click **Start** and select **Run** to open the Run utility.
- In the Run Open line type **D:\Setup.exe** to run the installation setup program (where D denotes the CD ROM drive in which the CD-ROM is inserted).
- To abort the installation without installing SmarTeam – MatrixOne Integration software, click **Exit**.

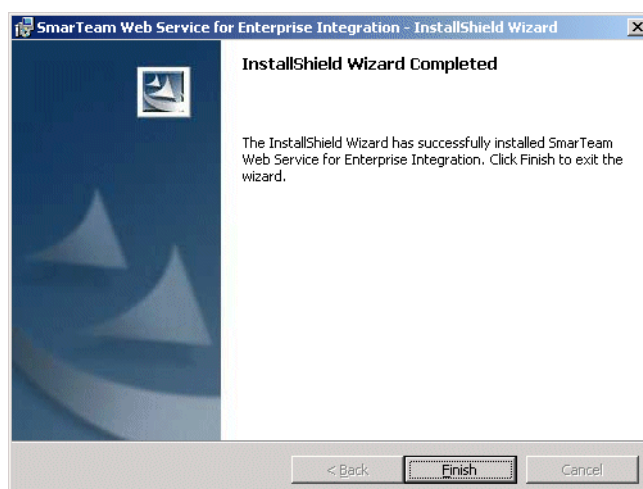
- 2 In the Welcome window, click **Next**.



- 3 In the Ready to Install window, click **Install**.



- 4 Click **Finish** to complete the installation and exit the InstallShield Wizard.



Chapter 4: Configuring SmarTeam – MatrixOne Integration

Configuring SmarTeam – MatrixOne Integration is described in this chapter. The following occurs:

- The system must be configured to connect the SmarTeam – MatrixOne Integration to SmarTeam. See [Connecting to SmarTeam](#).
- To enable mapping between SmarTeam users and classes to MatrixOne users and classes, configuration of the SmarTeam – MatrixOne Integration must occur. Mapping is an optional function. See [Mapping Users](#) and [Mapping Classes and Attributes](#).

Connecting to SmarTeam

To configure the SmarTeam – MatrixOne Integration so it can connect to SmarTeam, the following must be done:

- Configure IIS on the SmarTeam server on which the SmarTeam – MatrixOne Integration is installed.
- Authorize and authenticate the SmarTeam super user to view data.

A SmarTeam super user is given a certain set of permissions. SmarTeam verifies that the current user, or its mapped alias, exists in SmarTeam. Authentication of the current user is done at MatrixOne. The rights of a MatrixOne user are a subset of the rights of the SmarTeam super user.

For example, if a SmarTeam super user is unable to view a certain class, the MatrixOne user is not able to view it either. If the super user is configured to see classes X and Y, and the MatrixOne user is configured to see class Z, the MatrixOne user is not able to view any classes. If the same MatrixOne user is configured to see class X and not class Y and the super user is configured to see class X and class Y, the MatrixOne user is able to see class X only.

Configuring Users

From the System Configuration Editor, which is part of the SmarTeam Admin tools, the System Configuration is updated and the following keys are configured:

- `AdapletUserLogin`: Super username to connect to SmarTeam. Mandatory.
- `AdapletUserPassword`: Super user password. Mandatory.
- `smarTeamDatabaseReplicationId`: Replication ID used by the database to connect to SmarTeam. Optional. If a new value is provided, it overrides the Database Connection Manager default value.

The keys are stored in **`smarteam.std.MatrixIntegration.config.xml`**.

Mapping Users

The system administrator creates the **UserMapping.xml** file, which contains the mapping between SmarTeam users and MatrixOne current users. The SmarTeam – MatrixOne Integration hosts the **UserMapping.xml** file and uses it to map users.

For each operation, the following process occurs:

- MatrixOne sends a request containing the MatrixOne username to the SmarTeam – MatrixOne Integration.
- The SmarTeam – MatrixOne Integration checks if the MatrixOne username appears in the mapping.
- If the mapped username exists, it is used to connect to SmarTeam.
- If the mapped username does not exist, the SmarTeam – MatrixOne Integration uses the username sent by MatrixOne to connect to SmarTeam. In this case current users have the same usernames in SmarTeam and in MatrixOne.

Notes:

- Although mapping SmarTeam users with MatrixOne current users facilitates the retrieval of data, it is not required. The SmarTeam – MatrixOne Integration still enables MatrixOne users to request and retrieve data from SmarTeam even if the usernames are not mapped.
- If a MatrixOne user is trying to perform an operation, and is not recognized by SmarTeam, the MatrixOne Adaplet reads the usernames from SmarTeam again and refreshes the user mapping list to ensure that any user added to SmarTeam or added to the mapping since the Adaplet was initialized is recognized. This refresh operation occurs up to once a day as required.
- A [User Agent] must be mapped as a user in the system to enable the user to view see the metadata of mapped SmarTeam objects. The objects already appear according to current user authorizations.

Configuring UserMappingFile Path

To configure the UserMappingFile path, the following configuration key must be run:

- UserMappingFileLocation: Path for the user mapping file, which is called **UserMapping.xml**. The path is optional. However, if the path is provided, the xml file must also be provided.

The key is stored in `smarteam.std.MatrixIntegration.config.xml`.

The following is an example of a UserMapping file.

```
User Mapping XML
<?xml version="1.0" encoding="UTF-8" ?>
  <schema xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <User Smarteam_Name="joe" Matrix_Name="creator" />
  </schema>
```

Mapping Classes and Attributes

Users can map class names ([Class Mapping](#)) and attribute names ([Attribute Mapping](#)) between SmarTeam and MatrixOne.

Mapping is read only during initialization and is not refreshed until the service is restarted.

Class Mapping

To map a SmarTeam class to a MatrixOne class, the class must appear in the mapping file. A class can appear in the mapping file to display a SmarTeam class without mapping it to a MatrixOne class, although only selected attributes are displayed.

For example, a SmarTeam class name is displayed without MatrixOne symbolic and display names. Classes, which are not in the mapping file, appear as separate classes with all their attributes. They are not mapped to MatrixOne classes. If a SmarTeam class name does not appear in the mapping file, it is displayed as is. (See [Configuring MatrixOne](#).) Unmapped classes and their attributes are assigned MatrixOne names by the system.

If SmarTeam and MatrixOne classes are identical, they appear once in the MatrixOne database. Any query about the mapped class produces objects from the MatrixOne class and the SmarTeam class. To ensure that a class is not displayed by MatrixOne, the super user is not given view permission for this class.

The user must specify the SmarTeam class name, MatrixOne symbolic name and MatrixOne display name for the class.

Note: There is no mapping for linked classes. Objects related to SmarTeam objects are displayed by general or other kinds of links.

Attribute Mapping

A user can either display specific attributes by including them in the mapping file or hide specific attributes by excluding them from the mapping file.

MatrixOne attributes and mapped attributes should be of the same type.

Basic attributes can only be mapped at the root class level (except for name, which can be mapped at any level of the hierarchy). If mapping is not provided, default mapping is assigned.

If a user wants to display all attributes of the SmarTeam class, an XML attribute called **Attributes** is included with an **All** value. The attributes are displayed with the same names as in SmarTeam. (See the example of a schemaMapping file.) However, when mapping classes, the SmarTeam mapped class displays only a subset of the Attributes MatrixOne class. SmarTeam cannot display attributes that do not exist in MatrixOne. Therefore, the **All** value must be used carefully.

Configuring schemaMappingFileLocation

Mapping classes is handled via the **SchemaMapping.xml** file. To configure the schemaMappingFile path, the following configuration key must be run:

- **schemaMappingFileLocation**: Path for the schema file, which is called **SchemaMapping.xml**. The path is optional. However, if the path is provided, the xml file must also be provided. The SchemaMapping.xml file is hosted on the SmarTeam – MatrixOne Integration.

The key is stored in `smarteam.std.MatrixIntegration.config.xml`.

The following is an example of a schemaMapping file.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <schema xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
- <class Smarteam_Name="Projects" Matrix_Symbolic_Name="type_Products" Matrix_Name="Products">
  <attribute Smarteam_Name="TDM_RISK" Matrix_Name="attribute_CurrentVersion" />
  <attribute Smarteam_Name="TDM_DESCRIPTION" Matrix_Name="attribute_MarketingName" />
  <attribute Smarteam_Name="TDM_ID" Matrix_Name="attribute_Originator" />
  <attribute Smarteam_Name="TDM_START_DATE" Matrix_Name="attribute_StartEffectivity" />
  <attribute Smarteam_Name="STATE" Matrix_Name="attribute_Notes" />
  <attribute Smarteam_Name="MODIFICATION_DATE" Matrix_Name="attribute_StartEffectivity" />
  <attribute Smarteam_Name="CREATION_DATE" Matrix_Name="attribute_StartEffectivity" />
  <attribute Smarteam_Name="USER_OBJECT_ID" Matrix_Name="attribute_Synopsis" />
</class>
<class Smarteam_Name="Documents" Matrix_Symbolic_Name="type_Docs" Matrix_Name="Documents" Attributes="All" />
</schema>
```

Chapter 5: Configuring MatrixOne

To ensure that the MatrixOne user can view SmarTeam data, the connection between MatrixOne interface, via the MatrixOne applet and SmarTeam – MatrixOne Integration must be configured. Configuring MatrixOne is described in this chapter.

System Software Requirements

- The main MatrixOne version is V6R2009. In order to run MatrixOne, install V6R2009 HF0.
- After installing the HotFix, deploy the Web application in Tomcat (run WAR utility).

Configuring the params.xml File

Note: The params.xml file, which is a parameter used in Vault creation, must be configured before adding the Vault. (See [Configuring MatrixOne](#).) The xml file can be renamed, as long as the name and path are included in the Vault creation command.

MatrixOne runs this file to configure the connection between MatrixOne and the SmarTeam – MatrixOne Integration to enable file viewing from MatrixOne.

The following is an example of a params.xml file.

Note: Some of the keys can be edited and some have fixed values.

```
<settings>
  <localUpdates value="true" />
  <remoteUpdates value="false" />
  <normalizeDates value="false" />
  <defaultType value="Unknown Type" />
  <defaultOwner value="Unknown User" />
  <defaultPolicy value="Unknown Policy" />
  <defaultRelationship value="Unknown Relationship" />
  - <provider>
    <!-- URL to service provider -->
    <serviceURL value="http://172.16.108.77/SmarTeamAdapletService/SmarTeamAdaplet.asmx" />
    <authType value="basic" />
    <authUsername value="qa\qauser" />
    <authPassword value="qauser" />
    <symbolicNames value="true" />
    <timeout value="1800" />
  - <server>
    <vaultPattern value="eService Production" />
    <!-- the code will look at mx environemnt for the path -->
    <!-- <schemaFile value="\${MATRIXHOME}/etc/wsaSampleSchema.xml" /> -->
    <typePattern value="*" />
    <namePattern value="*" />
    <revPattern value="*" />
  </server>
</provider>
</settings>
```

Where:

- localUpdates value must be set to **true**.
- remoteUpdates value must be set to **false**.
- normalizeDates value must be set to **false**.
- authType value must be set to **Basic**.
- symbolicNames value must be set to **true**.
- timeout value is set to **1800**.
- defaultOwner is assigned as a default user when a SmarTeam object, which is displayed via MatrixOne, has no value, or when the username in that field is not recognized by MatrixOne. The defaultOwner must be a valid MatrixOne username.
- defaultType value can be set to **Unknown Type**.
- defaultPolicy value can remain **Unknown Policy**.
- defaultRelationship value can remain **Unknown relationship**.
- serviceURL value is set to the location of SmarTeam – MatrixOne Integration, such as `http://localhost/SmarTeamAdapletService/SmarTeamAdaplet.asmx`.
- authUsername [Domain\User] is a Windows user on the server where SmarTeam – MatrixOne Integration is hosted. Used for authentication purposes.
- authPassword value is the password of the Windows user on the server where SmarTeam – MatrixOne Integration is hosted.

Configuring MatrixOne

The following is handled by the MatrixOne administrator:

- 1 Run the MatrixOne MQL command utility.
- 2 Create and run the Add Vaults [name] command, which contains the params.XML file path. The following is the mandatory syntax for adding a Vault:
add vault [Vault Name] external file [params file name and path];

Example: add vault SmarTeam external file c:\matrix\params.xml;

- 3 After creating the vault, register it in MatrixOne.
- 4 Add the Vault as a secondary Vault in the relevant company [SmarTeam].
- 5 Create a user with [Unknown User] username.
- 6 Open Business application and search for **IEFShowVersionLabel** program. Change the following:
result!=null && results()>=0 to **>0**
- 7 To view SmarTeam classes as they appear in SmarTeam via MatrixOne, do the following:
 - a Stop **Apache**.
 - b Modify the emxSystem.Properties file by adding the **type_SmarTeamTypes** value to the emxFramework.GenericSearch.Types key in the following locations:
 - C:\Program Files\Apache Software Foundation\Tomcat 5.5\webapps\ematrix\WEB-INF\classes

- C:\eMatrix\distrib\ematrix\WEB-INF\classes
- c** Modify the emxEngineeringCentral.properties by adding the **type_SmarTeamTypes** value to the eServiceEngineeringCentral.Types key in the same two locations.
- d** Open the Business application and search for the object **ENCSearchGeneral** of type **Command**. Add **type_SmarTeamTypes** to the **Href** field.
- e** Open MQL and run the following command: add store StStore type external vault [Vault Name] file [params file name and path]
- 8** Open Business application and search for Documents Policy. Add **generic** to Allowed Formats.
- 9** Create SmarTeam Configuration object to show search results properly in the Designer Central.
 - a** In the Matrix application, create a new object with **MCADInteg-GlobalConfig** type and **smarteam** name.
 - b** In the Attributes window type the values for the following attributes:

Attribute name	Attribute value
MCADInteg-RelationshipClassMapping	CADSubcomponent AssemblyLike
MCADInteg-TypePolicyMapping	SmarTeam Types Documents Policy
MCADInteg-RelMapping	CAD Subcomponent to, CAD SubComponent
MCADInteg-BusTypeMapping	smarteam SmarTeam Types
IEF-ReleasedState	Documents Policy New
IEF-Pref-MCADInteg-DefaultTypePolicySettings	(DEFAULTVALUE)SmarTeam Types Documents Policy,Versioned Design Policy
IEF-FinalizationState	Documents Policy New
MCADInteg-TypeClassMapping	TYPE_CADMODEL_LIKE smarteam,assembly,component TYPE_COMPONENT_LIKE smarteam,component TYPE_ASSEMBLY_LIKE smarteam,assembly
MCADInteg-TypeFormatMapping	smarteam SmarTeam Types,generic

- c** Open MQL and run the following command: execute program IEFRegisterIntegration - method registerIntegrationToIEF smarteam 10.7.1 [date <2/7/08>] smarteam;
- d** Refer to DSC Admin Guide to assign Integration to the user.
- e** Restart the Apache server and reload the cache.

Chapter 6: Working with SmarTeam – MatrixOne Integration

After SmarTeam – MatrixOne Integration is installed and configured, and MatrixOne is configured, the SmarTeam – MatrixOne Integration is connected. Now, the MatrixOne user is ready to view SmarTeam data. Working with SmarTeam – MatrixOne Integration is described in this chapter.

Working with SmarTeam – MatrixOne Integration

The authenticated MatrixOne current user is only allowed to access data according to the super user's authorizations in SmarTeam (via the Security module). The following conditions exist while the MatrixOne current user is working with SmarTeam – MatrixOne Integration:

- 1 The MatrixOne current user can search objects from the SmarTeam database and explore them in the Designer Central as well as view files with the native application, such as a *.doc file, which is opened using Microsoft® Office and a CATIA part, which is opened in CATIA.
- 2 The updated SmarTeam object's properties appear in the Designer Central.
- 3 If the metadata is updated in SmarTeam, the last modification appears in the Designer Central after performing a Refresh operation [F5].
- 4 The SmarTeam object can be referenced to a MatrixOne object. For example, the SmarTeam document can be added to the MatrixOne collection.
- 5 The SmarTeam linked objects can appear in the PowerView of the Designer Central.
- 6 Navigating within the CAD Assembly or BOM Items is performed in the Designer Central.

Limitations

The following limitations exist in the system:

- If a class in SmarTeam is mapped to a class in MatrixOne, it cannot contain attributes that do not exist in MatrixOne.
- Mapped attributes must be of the same type.
- Regardless of the class and attributes mapping, The following can restrict the metadata displayed to MatrixOne:
 - Metadata is displayed according to super user authorizations.
- **Find in Files** is not supported by SmarTeam.
- Query **Search > Engineering central > Find Related** is not supported.
- **Query for Object** with conditions on its links' attributes is not supported.

Chapter 7: Troubleshooting

Known Issues

For installation known issues, refer to <http://support.smarteam.com/>.

Frequently Asked Questions

For installation frequently asked questions, refer to <http://support.smarteam.com/>.