



ENOVIA SmarTeam

System Configuration

© Dassault Systèmes, 2004, 2009. 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 Number: GEN-V1-190409

Table of Contents

CHAPTER 1: INTRODUCTION	1
Overview	1
Upgrading from Previous SmarTeam Versions	1
Migrating from V5R10 and V5R11	1
Migrating from V5R12 and V5R13	1
SmarTeam Corporation Internet Site	1
Related Documentation	2
CHAPTER 2: TERMINOLOGY	3
Configuration Key (Element)	3
Configuration Set (Header)	3
Configuration Files	3
Configuration Level	5
Example	6
Configuration Key Hierarchy	8
Configuration Key Summary	8
System Configuration Override Levels	9
CHAPTER 3: USING THE SYSTEM CONFIGURATION EDITOR	12
Entering the System Configuration Editor	12
System Configuration Main Page	12
View Types	12
Filtered View	13
Unfiltered View	14
Adding a New Key Value	15
Deleting a Key Value	15
Editing a Key Value	15
CHAPTER 4: IMPLEMENTATION	16
Key Types	16
Connecting SmarTeam Applications to System Configuration Service through a Firewall	16
Mapping Pre-V5R13 Repositories	16
Transporting System Configuration	18
General	18
Domain Level	19
User Level	20
Adding Complex Keys to the System Configuration Service	23
Adding a New Key to the System Configuration	26
Writing to the System Configuration Service using SmarTeam API (session smConfig)	26
Defining the Default Connection for the SmarTeam Database for Multiple Clients	27
Using Individual Configurations for Different Users	27
Manually Editing System Configuration XML Files	27
System Configuration Service	27
Configuration Schema	27
Configuring the System Configuration Service to Work with the Windows Protocol	28
Configuring Core Services on a Multi-Network Card Machine	29
Configuring IIS 64bit to Work with an Application running at 32bit	30
XML Troubleshooting	30
All XML elements must have a closing tag	30
XML tags are case sensitive	30

All XML elements must be properly nested	30
All XML documents must have a root element	31
Attribute values must always be quoted	32
With XML, white space is preserved.....	32
Comments in XML	32
CHAPTER 5: CONFIGURING SESSION MANAGEMENT.....	33

Chapter 1: Introduction

Overview

The System Configuration Service provides a centralized mechanism that contains all configuration-related information for all SmarTeam applications.

The system configuration service has multiple levels of configuration allowing easier manageability and security across sites, machines, applications, databases and users from anywhere in the organization.

The administration of the System Configuration Service is performed using a web-based application.

This guide provides a description of the functionality and administration of the SmarTeam System Configuration Service.

Upgrading from Previous SmarTeam Versions

If you are upgrading from a previous version of SmarTeam, you need to run the SmarTeam System Configuration Migration Wizard. For details, refer to the SmarTeam Procedure for Upgrading to V5Rxx.

Migrating from V5R10 and V5R11

When upgrading from V5R10 and V5R11, all keys are migrated directly to the new System Configuration System.

Migrating from V5R12 and V5R13

There is a conceptual difference between SmarTeam R12 and R13 (and above) regarding the System Configuration Service:

- In R12, the System Configuration Service was not integrated into the entire SmarTeam system, it was only used by the Session Management service, NLS and Help sub-systems.
- From R13, the System Configuration was fully integrated in SmarTeam. All configurations that were previously saved in the database, .INI files and registry were merged into the System Configuration Service.

SmarTeam Corporation Internet Site

You are highly recommended to frequently visit the SmarTeam Corporation Internet site for the latest updates to SmarTeam and plug-in products, including the latest Service Packs, hot fixes and technical support at <http://www.smarteam.com>.

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

Related Documentation

The following documentation is referred to in this guide.

Name of Document	CD Location	Remarks
SmarTeam Procedure for Upgrading to V5R17.pdf	SmarTeam Program Directory	Details the upgrade procedure if you are upgrading from a previous version of SmarTeam – Web Editor.
V5R17 Hardware and Software Requirements.pdf	SmarTeam Program Directory	Details the hardware and software required for a successful installation.
SmarTeam – Multi-site Administration Guide.pdf	SmarTeam Documentation Directory	Details the various administration procedures to successfully set up, customize and maintain a SmarTeam – Multi-site system in a corporate environment.

Chapter 2: Terminology

Configuration Key (Element)

Configuration keys define the parameters in the SmarTeam applications.

Configuration Set (Header)

A configuration set is a collection of configuration keys. The configuration set represents the transactional unit for the System Configuration Service. Each configuration set is defined by a unique schema.

Note: In R10, the term “configuration set” was known as “header”

In general, a configuration set represents a set of configuration elements that are logically grouped together. For example, the configuration set "smarteam.std.lifeCycle.config" holds keys related to the configuration elements of the lifecycle.

Configuration Files

A configuration file is a physical collection of keys on the computer's disk. Each configuration file contains keys, not necessarily related to its purpose. For example, keys related to lifecycle can appear in the configuration files smarteam.std.legacyPreferences and in smarteam.std.lifeCycle. Each configuration file has a unique name.

The following configuration files currently appear in the SmarTeam system:

Name	Description
smarteam.std.legacyPreferences	Contains configuration keys that was migrated from SmarTeam .INI files, Registry and Database
smarteam.std.clientLibraries	Contains the entire configuration for the client libraries sub-system.
smarteam.std.configClient	Contains configuration keys related to the System Configuration Client
smarteam.std.configurationManagement	Contains configuration keys related to the SmarTeam Configuration Management sub-system
smarteam.std.dynamicTypeMappings	Contains the mapping between user-defined types and their strong name types. This file is related to SmarTeam – Web Editor.
smarteam.std.embeddedScripts	Contains configuration keys related to the Embedded Scripts Service
smarteam.std.externalApplications	Contains configuration keys related to the add-ons of SmarTeam – Web Editor.
smarteam.std.FavoriteSearches	Contains configuration keys related to the favorite searches in SmarTeam – Web Editor.
smarteam.std.FileStorageManager	Contains configuration keys related to the File Storage Manager sub system.

Name	Description
smarteam.std.Foundation	Contains configuration keys related to the Foundation module of the SmarTeam – Web Editor.
smarteam.std.Help	Contains configuration keys related to the Help sub-system.
smarteam.std.Integration	Contains configuration keys related to SmarTeam – Web Editor’s support for integrations.
smarteam.std.lifeCycle	Contains configuration keys related to the Life Cycle sub-system.
smarteam.std.messages	Contains configuration keys related to the messaging support in the SmarTeam – Workflow sub-system.
smarteam.std.nls	Contains configuration keys related to the NLS sub-system.
smarteam.std.nlsTranslationTool	Contains configuration keys related to the NLS Translation Tool
smarteam.std.persistencyStorage	Contains configuration keys related to the Persistency Storage sub-system.
smarteam.std.preferences	Contains general preferences related to SmarTeam – Web Editor.
smarteam.std.programManagement	Contains configuration keys related to the Program Management sub-system.
smarteam.std.progressIndicator	Contains configuration keys related to the progress indicator sub-system.
smarteam.std.redirect	Contains configuration keys related to the SmarTeam Redirect module.
smarteam.std.viewers	Contains configuration keys related to the different viewers
smarteam.std.views	Contains configuration keys related to Views support in SmarTeam – Web Editor.
smarteam.std.webControls	Contains configuration keys related to the SmarTeam Web Controls
smarteam.std.webEditor	Contains configuration keys for SmarTeam – Web Editor.
smarteam.std.Workflow	Contains configuration keys for the Work Flow sub system
smarteam.std.Authentication	Contains configuration keys related to the Authentication Modules used in the Session Management service.

Configuration Level

The System Configuration Level is a logical collection of System Configuration keys. These keys become the default keys for an end-user if they are not overridden on the lower levels.

The keys are not linked functionally, but are simply used to set up the defaults for the lower levels.

The System Configuration Service supports multiple configuration override levels to allow the definition of complex configuration combinations. The concept behind the configuration levels is the ability to allow information to be added or changed in the configuration according to the appropriate level.

In each level, values can be assigned to keys that did not have *or have different* values in the previous levels.

If a key has different values in two different override levels, the value belonging to the lower level is used. A sample screen presenting override levels is shown below:



The following table presents the available configuration override levels:

Configuration Override Level	Hierarchy	Description	Location
Default	0	The default values are used for all levels. If no value is defined on the lower level – the default value will be taken. The default values are read-only.	<SmarTeam>\ConfigurationSettings\Default
Domain	1	The Domain level is the topmost level. This level defines the default configuration for all applications, machines, system and users. Note: There can be several systems in the domain	<SmarTeam>\ConfigurationSettings\Domain

System (Database)	2	The System level controls the configuration per system id. A system id represents any system that SmarTeam applications can connect to (such as a relational database). For example, a system id can comprise a database replication ID.	<SmarTeam>\ConfigurationSettings\<systemid>
Site	3	The Site level is used to allow different configurations for different sites.	<SmarTeam>\ConfigurationSettings\Site
Machine	4	The Machine level controls the configuration per machine (identified by machine name).	<SmarTeam>\ConfigurationSettings\<machine-name>
Application	5	The Application level controls the configuration per application (identified by name). For example: SmarTeam – Editor, SmarTeam – Web Editor, SmarTeam – Community Workspace.	<SmarTeam>\ConfigurationSettings\<application-name>
User	6	The User level controls the configuration per user (identified by a unique text string). For example: joe. Note: By default, this mostly concerns Visual Settings	<SmarTeam>\ConfigurationSettings\<user-strong-name>

Example

The following example explains how the override levels work.

A new key is added on the system level:

The screenshot shows the SmarTeam configuration interface. On the left is a navigation tree with categories like 'Miscellaneous Configuration', 'Conversion Formats', 'Server Hooks', 'Scripts Preferences', 'Directory Structure', 'General Preferences', 'Life Cycle Preferences', 'Messages Preferences', 'Options', 'Vault Preferences', 'Workflow', 'User Account Management Preferences', 'Design Copy', 'National Languages Support', 'Help Preferences', 'Collaborative Design', and 'Applications'. The 'Scripts Preferences' section is expanded. The main panel displays configuration details for 'smarteam.std.legacyPreferences.config'. The 'Key Name' is 'Directory_Structure.ScriptDirectory'. The 'Override Level' is set to 'System'. A yellow circle highlights the 'Please select a system:' dropdown, which shows the value 'smarteam://5FE013157E6D11D5B95F0004AC586F73'. A speech bubble points to this dropdown with the text: 'This definition is done on the system level for the Database with the DB ID 5FE013157E6D11D5B95F0004AC586F73. The correct format for the override form on this level is **smarteam://DB ID**'. The 'Value' field contains 'C:/Scripts'.

This definition is shown in the Override Form field for this key:

The screenshot shows the Smarteam configuration interface. On the left is a tree view under 'Miscellaneous Configuration' with various sub-items. The main area displays configuration details for 'smarteam.std.legacyPreferences.config'. The 'Key Name' is 'Directory_Structure.ScriptDirectory'. The 'Override Level' is 'System', and the 'Override Form' is 'smarteam://5FE013157E6D11D5B95F0004AC586F73'. The 'Value' field contains 'C:\Scripts'. A yellow oval highlights the 'Override Level' and 'Override Form' fields.

View Type:	Filtered View (Grouped)
Configuration Set Name:	smarteam.std.legacyPreferences.config
Key Name:	Directory_Structure.ScriptDirectory
Override Level:	System
Override Form:	smarteam://5FE013157E6D11D5B95F0004AC586F73
Value:	C:\Scripts

As shown below, a user connected to the database with DB ID 5FE013157E6D11D5B95F0004AC586F73 will be using the script directory C:\Scripts.

For users connected to any other database, the scripts directory "Home Directory\Script" will be used.

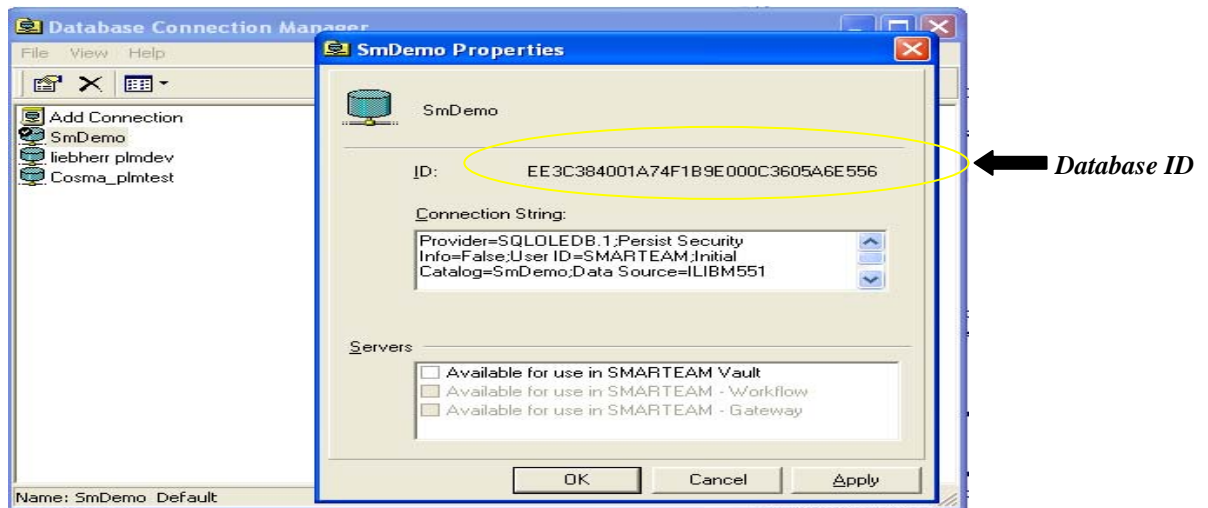
The screenshot shows the Smarteam configuration interface with a table of override values. The table has columns for 'Value', 'Override Level', and 'Override Form'. There are three entries: '<HomeDirectory>\script' with 'Default' level, '<HomeDirectory>\script' with 'Domain' level, and 'C:\Scripts' with 'System' level. The 'System' level entry has the 'Override Form' 'smarteam://5FE013157E6D11D5B95F0004AC586F73'. Buttons for 'Add Value' and 'Delete Value' are visible.

Value	Override Level	Override Form
<input type="checkbox"/> <HomeDirectory>\script	Default	
<input type="checkbox"/> <HomeDirectory>\script	Domain	
<input type="checkbox"/> C:\Scripts	System	smarteam://5FE013157E6D11D5B95F0004AC586F73

To obtain the database ID:

- Run the Database Connection Manager and search for the entry smarteam.std.legacyPreferences.config in the System Configuration Editor.
- or:
- Run the Database Connection Manager, select the database and right-click on Properties.

The following screen appears:



The Database ID is shown at the top of the window.

Note: When working in the SmarTeam – Multi-site environment, the script directory is defined on the Site level instead of System level, so DB_REPLICID is used instead of DATABASE_ID.

Configuration Key Hierarchy

Configuration Key Summary

The Configuration Key Summary is a logical collection of the final settings of all keys that are applied for a specific user.

For example: In the table below, key 1 has been assigned values at the Domain, System and User levels.

	Key 1	Key 2	Key 3	Level
	1	1	1	Domain Level
	2		2	System Level
	3			User Level
TOTAL	3	1	2	

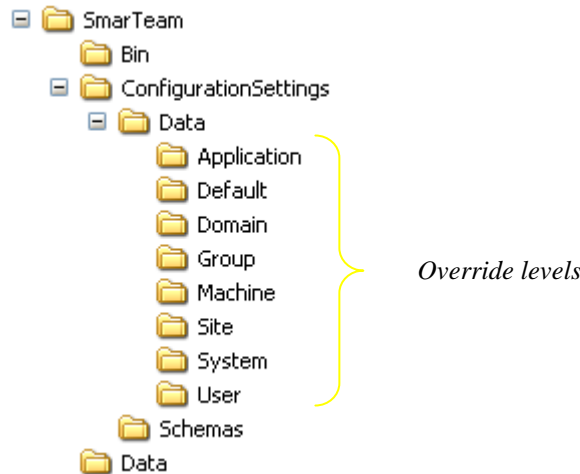
The final value for key1 is 3 as the override level for Users takes precedence.

The final value for key2 is 1 as only the Domain level exists has a value for this key.

The final value for key3 is 2 as the override level for Systems takes precedence.

System Configuration Override Levels

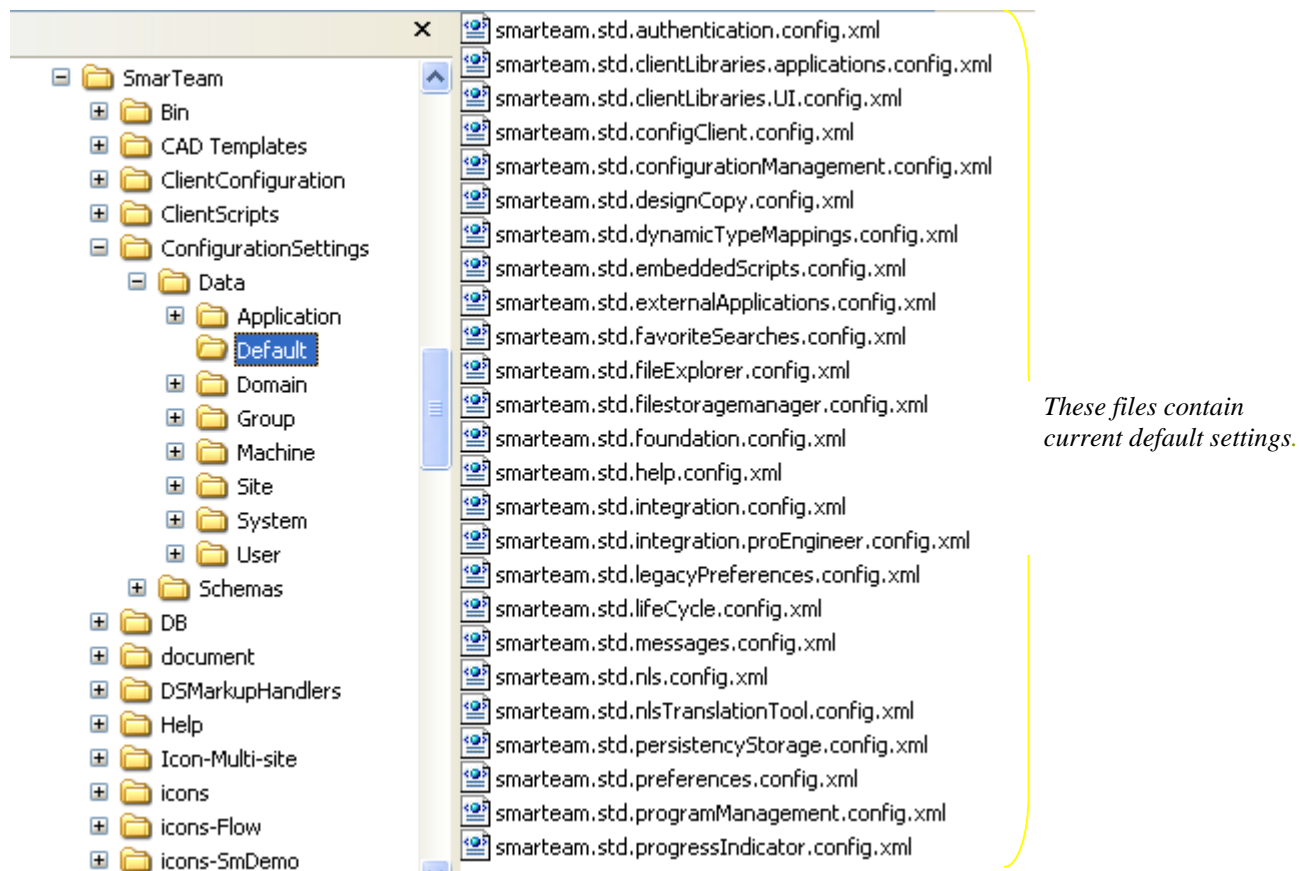
The following diagram shows a representation of the override levels contained in the System Configuration Service. The files are located under <SmarTeam Home Directory>\ConfigurationSettings\Data as shown below:



Default Level

The settings defined at the Default level are located under the Default folder.

The differences between settings for each version are stored in the appropriate folder under the Default folder as shown below:



Domain Level

The settings defined at the Domain level are stored under the Domain folder.

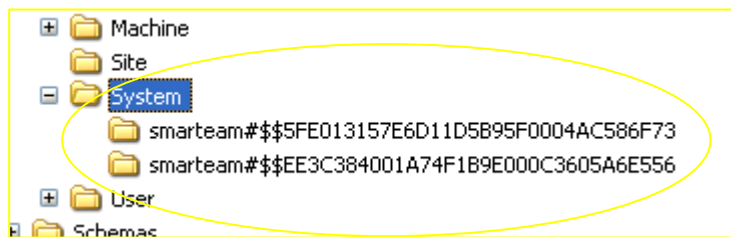
System Level

The settings defined at the System level are stored under the System folder. For each database, a sub-folder is created. The name of the sub-folder is in the following format:

smarteam#\$\$<datatabaselD>

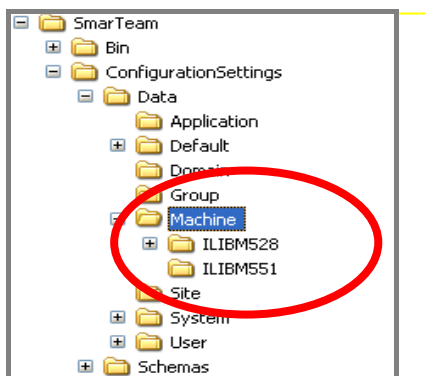
For example: **smarteam#\$\$5FE013157E6D11D5B95F0004AC586F73**

A sample System folder is shown below:



Machine Level

The settings defined at the Machine level are stored under the Machine folder. For each machine, a sub-folder is created in which the modified xml files are stored. A sample Machine folder is shown below:



User Level

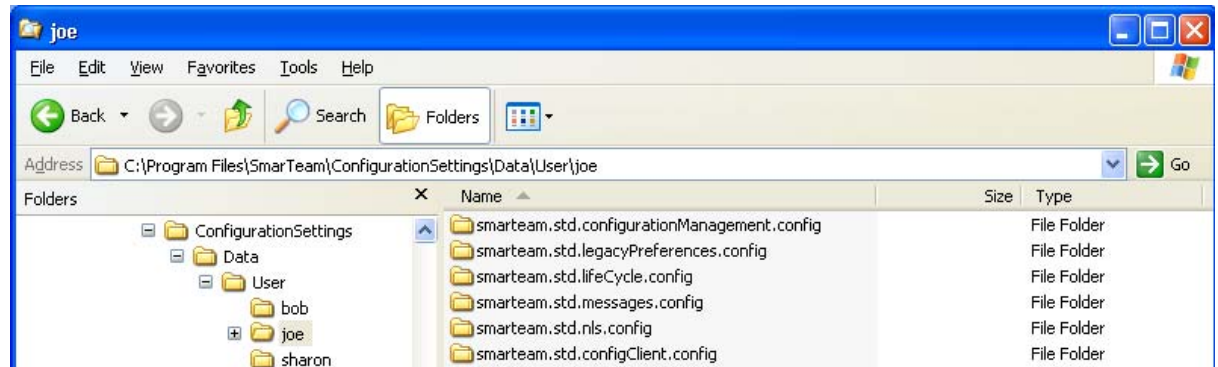
The settings defined at the User level are stored under the User folder. For each user for whom specific settings have been defined, a new sub-folder is created. The name of the sub-folder is in the following format:

<user login name>

For example: **bob**

XML files that have been specifically modified for this user are shown in the User folder.

A sample of a User folder and its contents is shown below.



Site Level

From R17, you can add / change configuration settings at the Site level. For example, when upgrading from R14 to R17, you can add configuration settings in a Test environment. All settings that are only relevant for the Test environment and should not be replicated in the Production environment should be put into the Site level.

For example:

The key `SmarTeam.Database_Connection_Setup.1.Database_Connection_String` can be changed in the `legacyPreferences.xml` file under the Site Level in a Test environment in order to test the setup, but these changes will not be replicated to the Production environment.

In Multi-site environments, if there are individual configuration settings on each site, this method will work correctly. However, you cannot use a single configuration setting for more than one site in a SmarterTeam – Multi-site environment.

Application Level

Not currently implemented.

Chapter 3: Using the System Configuration Editor

The System Configuration Editor is an administrative tool that is accessible to administrators only. To access the System Configuration Editor, the user needs to provide authentication by supplying the user name and password. When using the SmarTeam authentication protocol and when there is no specific database for authentication defined, the System Configuration Editor Login page will also enable the user to select the database to use when authenticating the user name and password.

Entering the System Configuration Editor

To enter the System Configuration Editor:

1. From the Start Menu, select **Programs > SmarTeam > Administrative Tools > System Configuration Editor**

Or

1. Go to the following URL:

`http://<ComputerName>/SmarTeam/System/ConfigurationEditor/`

where ComputerName is the name of the machine on which the utility is installed.

Note: When logging into the System Configuration Editor, the database shown as selected may not be the one defined as default using the Database Connection Manager. Select the correct database from the list.

System Configuration Main Page

The System Configuration Service Main Page is divided into two areas:

Left panel: contains a list or tree of the configuration sets

Right panel: Contains configuration keys and their values.

View Types

There are different ways of viewing information in the System Configuration Service. These are known as "views".

To change the current view type:

1. Click the View Type combo box in the left panel.
2. A list of view types appears as listed below:
 - **Filtered View:** The configuration sets are represented as a hierarchical structure sorted into different groups. These groups can be customized according to your requirements. Note that configuration sets that were not added to a group will not be seen in this view.
 - **Unfiltered View:** All configuration sets that exist in the System Configuration Service are shown even if they were not added to the "Filtered View".
 - **Filter by Site:** The configuration sets for which a certain site has override values are shown.

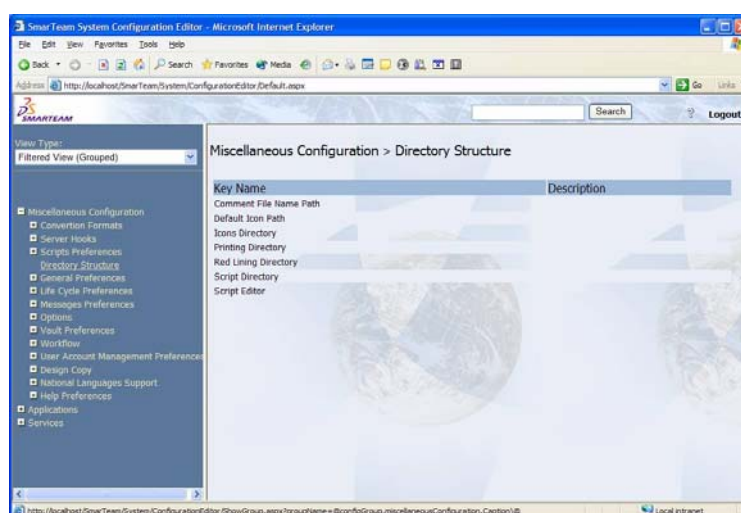
- **Filter by Machine:** The configuration sets for which a certain machine has override values are shown.
- **Filter by Application:** The configuration sets for which a certain application has override values are shown.
- **Filter by System:** The configuration sets for which a certain system id has override values are shown.
- **Filter by User:** The configuration sets for which a certain user has override values are shown.

Filtered View

In the Filtered View the configuration sets are represented in a hierarchical structure for ease of use. The configuration sets are sorted into different groups.

Note: These groups can be customized according to your requirements.

A sample filtered view is presented in the left panel below:

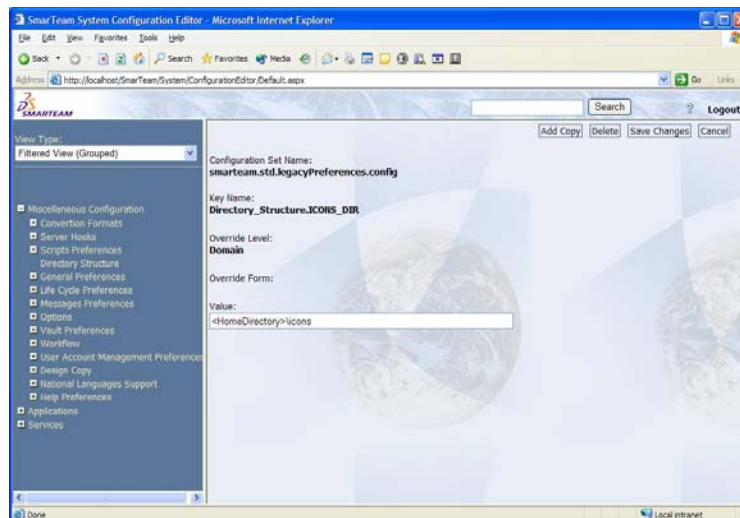


To view the keys for a specific configuration set:

1. Click on the Configuration Set Name.

A list of available override levels appear.

2. Click on an override level to view its override keys in the right panel.
3. Click on a key name to view its values, as shown in the sample screen below.



From this page you can add a new key or delete an existing key. See [Adding a New Key Value](#), [Deleting a Key Value](#) and [Editing a Key Value](#) for details.

Unfiltered View

In the Unfiltered View the configuration sets are presented in a running list in the left panel.

To view the keys for a specific configuration set:

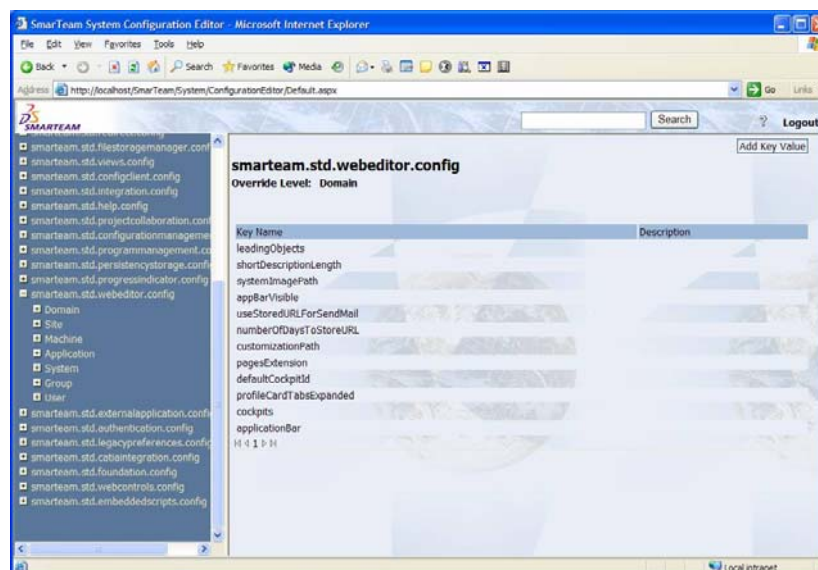
1. Click on the Configuration Set Name.

A list of domains appears.

2. Click on a domain to view its override keys in the right panel.
3. Click on a key name to view its values, as shown in the sample screen above.

From this page you can add a new key or delete an existing key value. See [Adding a New Key Value](#), [Deleting a Key Value](#) and [Editing a Key Value](#) for details.

A sample unfiltered view is presented in the screen below:



In addition to the Filtered and Unfiltered Views, you can also view the configuration sets according to Site, Machine, System, Application and User.

Adding a New Key Value

From each view type, a new key value can be added.

To add a new key value:

1. Click Add Key Value.

The Add Key Value window appears.

2. Complete the fields as follows:
 - Value: Enter the value for the key.
 - Override Level: The desired override level for which this value is inserted
 - Override Form: The override form (depending on the override level) for which this value is relevant. For example, when entering a value that is relevant to a certain machine, the override form will contain its name.

For example, in the above screen, for a Time Stamp, a value of TestApp at the Application level overrides any values that appear at the System level or lower.

Deleting a Key Value

From each view type, an existing key value can be deleted.

To delete a key value:

1. Click the checkbox next to the key name.
2. Click Delete Key Value.

The key value is removed from the System Configuration Service.

Editing a Key Value

You can update the value of a key at any time.

To update a key value:

1. Click on the Value field to display the values available for this key.
2. Click on the desired value.
3. Click Save Changes.

The key value is updated.

Chapter 4: Implementation

Key Types

There are three types of keys in the System Configuration Service:

- **Static Key:** System keys that are optimized for read-only operations. Using these keys for read/write operations results in a tax of system resources.
- **Dynamic Key:** A specific type of user-defined key that is dynamically and automatically changed during a client's work. This type of key is normally defined during the customization process.
- **User-defined Key:** A key that is specifically defined by the user for customization purposes.

Note: When adding a user-defined key, it must be saved in a local ini file on the client machine.

Connecting SmarTeam Applications to System Configuration Service through a Firewall

This section is related to the connection of SmarTeam applications to the System Configuration Service through Firewalls.

The System Configuration Service is a communication module implemented using Microsoft®'s Remoting.NET / SOAP Web Service technology. By implementing the System Configuration Service with Remoting.NET / SOAP Web Service, the System Configuration Service can work on top of any Remoting.NET-supported protocol. By default, the supported protocols are TCP and HTTP.

After installation, the default settings for the System Configuration Service are the protocol TCP on port 5607.

To change the default settings of the service, open the **SystemConfigurationRemoting.config** file located in SmarTeam's Bin directory on the machine on which the System Configuration Service was installed.

Mapping Pre-V5R13 Repositories

Configuration information that was previously stored in .INI files, registry and database preferences is now part of the System Configuration Service.

All types of keys that were inherited from previous SmarTeam versions are currently located in the **smarteam.std.legacypreferences.config** configuration set.

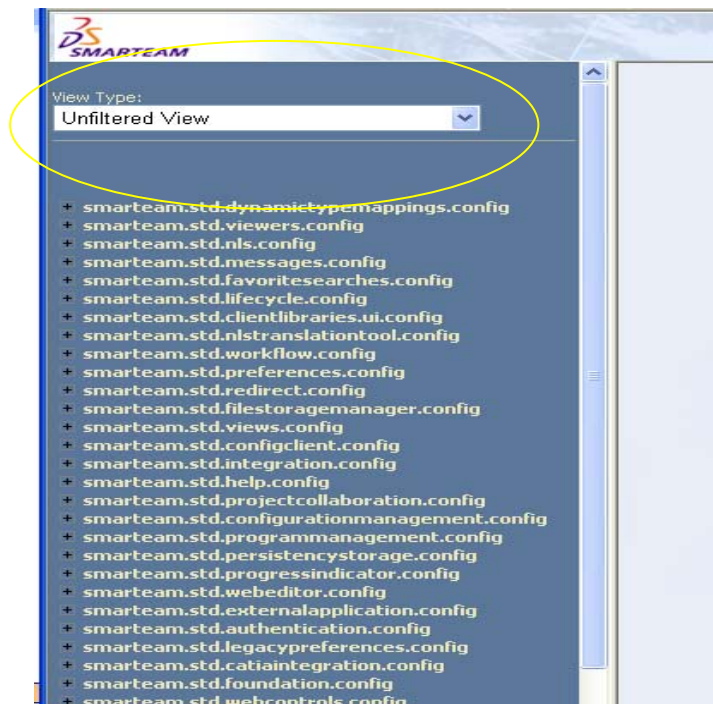
The System Configuration Service includes a search facility that allows users to find keys by their names. The search finds all elements whose name matches exactly or contains the specified text.

The search facility presents a list of all corresponding configuration elements and the configuration set to which they belong.

Clicking on the key name redirects you to the key view which shows all available values for this key.

In addition, you can use the information described in the table below.

Note: To view all keys of a certain configuration set, you must work in Unfiltered view. A sample Unfiltered view is shown below:



The following table describes the method of mapping configuration data for all sources (.INI, Registry, database) to the System Configuration Service. The left column contains samples of entries in the old .INI files, Registry and Database Preferences. The right column contains the location and entries in the System Configuration Service that replaces these.

Configuration Location (PREVIOUS)	Mapping in System Configuration (CURRENT)
.INI Files	
Format: [SectionName] KeyName=Value Example: [Directory_Structure] ScriptDirectory=<HOMEDIRECTORY>\script	Format: SectionName.KeyName This parameter is found under the Configuration Set: smarteam.std.legacypreferences.config under the Key Name: Directory_Structure.ScriptDirectory
Registry	
Example: \$Admin\Database Connection Info\0\Database ID	This entry is found under the Configuration Set: smarteam.std.legacypreferences.config under the Key Name: Database_Connection_Info.0.Database_ID

Configuration Location (PREVIOUS)	Mapping in System Configuration (CURRENT)
Preferences (Inside the database)	
Example: CONVERSION_FORMATS.Date	Example: This entry is found under the Configuration Set: smarteam.std.legacypreferences.config under the Key Name: CONVERSION_FORMATS.Date

Transporting System Configuration

General

An administrator can prepare system configuration in one environment and then apply it safely to another environment. For example, you can create system configuration in the Test environment and then apply it in the Production environment.

System Configuration files created in one SmarTeam system, contain a database identifier that needs to be changed while moving to another SmarTeam system.

The database identifier of the source system should be replaced with the database identifier of the target system.

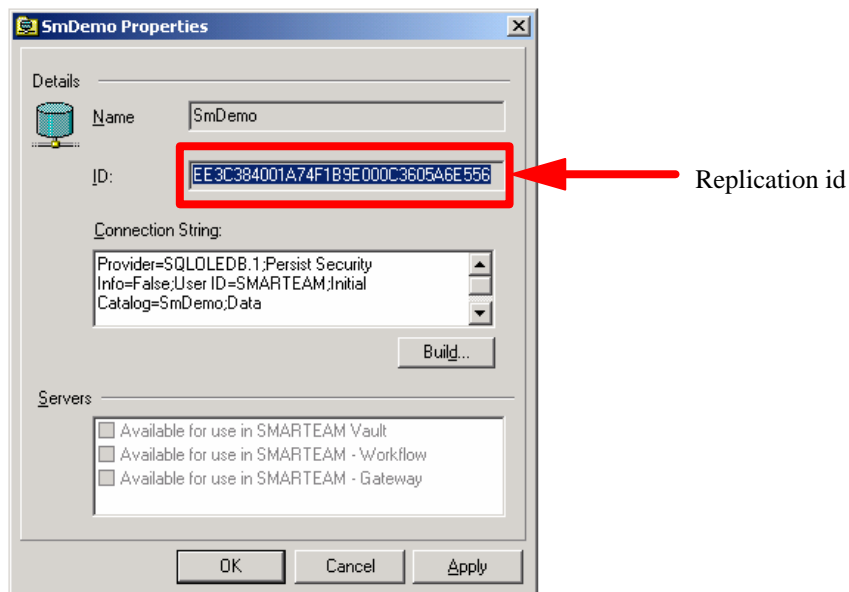
All entries in the System Configuration that are directly linked to the source environment elements, for example, computer names, IP addresses etc., should be replaced with appropriate data for the production environment.

To transport system configuration schema:

1. Take the configuration files from the source server.
2. Copy them to the appropriate location on the target server.
3. Change all entries in the configuration files of the target configuration from source database DATABASE_ID (select DATABASE_ID from TDM_DB_VERSION) to the target database DATABASE_ID.

Note: If you do not know the ID of the target database, you can copy it from the Database Properties window as follows:

- a. Go to the Database Connection Manager.
- b. Click on the required database. The Database Properties window appears.



c. Copy the database ID string from the ID field.

Domain Level

The main file to be changed at the domain level is:

<SmarTeam>\ConfigurationSettings\Data\Domain\smarteam.std.legacyPreferences.config.xml

Important: When migrating, before attempting to connect to the system, change the connection string. For full details on migration, refer to [Upgrading SmarTeam to V5R17.pdf](#).

In SmarTeam - Multi-site systems, this issue is related to a site where the System Configuration is copied to. After copying the files, the connection string must be changed for each site.

User Level

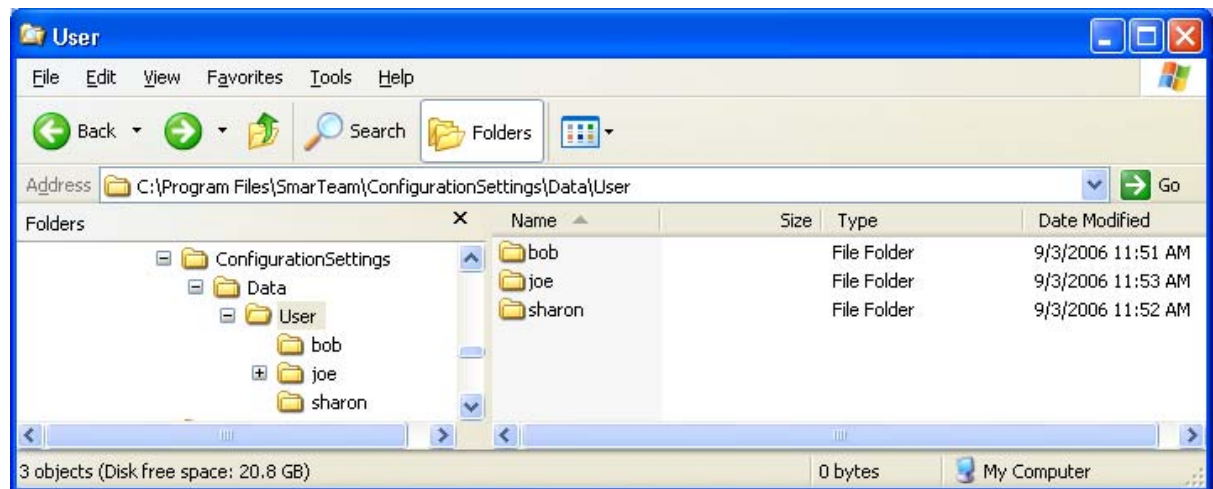
User-specific preferences can be created for one user and then distributed to other users. In order to do this, specific folders must be created inside the User System Configuration folder.

For example, if s there are two users - “Joe” and “Bob”, a folder named “joe” is created in the USER System Configuration folder when Joe logs in for the first time.

To distribute this folder to “Bob”:

1. After creating the necessary System Configuration settings for “Joe”, such as appropriate Visual settings, log out of SmarTeam.
2. Copy the folder named “joe” from the User System Configuration folder to the folder called “bob” (see the diagram below).
3. When “Bob” logs in for the first time, he will see the same Visual settings as Joe sees.

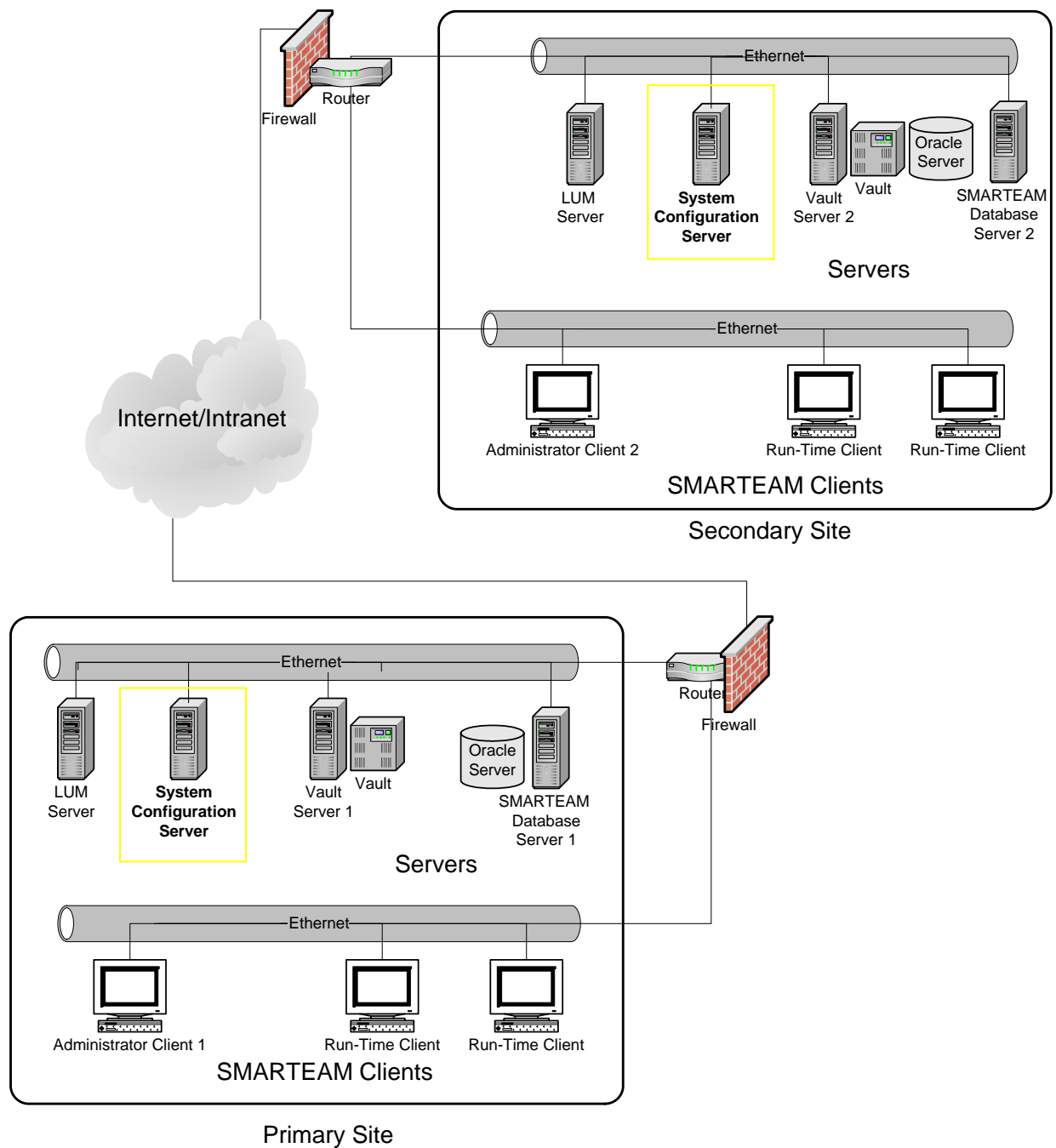
Note: If “Bob” has already logged in, even once, his own System configuration settings already exist. Therefore, in order to apply settings of “Joe” to “Bob”, you must first delete Bob’s settings (folder “bob”) and then replace the folder with the folder “joe”, as described above.



Managing System Configuration in the MUS Environment

The System Configuration in the MUS environment behaves as in a regular environment, except for a few small changes. For full details, refer to the SmarTeam – Multi-site Administration Guide.

The following diagram shows a typical Multi-site system:



SmarTeam – Multi-site Specific Changes

When transporting System Configuration in the SmarTeam – Multi-site environment, you can prepare System Configuration at one site and distribute the configuration to other sites.

In order that users can connect to their relevant sites, when transporting, the connection string in the System Configuration must be changed. This can be done in the System Configuration Editor.

The entire process of transporting System Configuration holds true also for the SmarTeam - Multi-site system. The only exception is DATABASE_ID, which is the same for all sites within the SmarTeam – Multi-site system.

Synchronization of System Configuration Files

When changing keys at one site, if you want to synchronize between sites, relevant files may be distributed to other sites or the same changes can be made once again at each site. This should be done when no users are connected. This is to avoid a situation in which sites have different settings for the same modules (e.g., Lifecycle), which may result in data corruption.

Note: It is strongly recommended that you locate the System Configuration service on a different server than the one on which SmarTeam is installed.

Sometimes there is a need to connect remote clients to a SmarTeam database and the MUS is not yet applicable. In that case, the recommendation is to place the System Configuration Server along with the NLS storage closer to the remote users. This may result in several System Configuration Servers related to the same database.

After using scheduled synchronization, note that in order to connect users to the correct vault, two parameters need to be synchronized manually:

- Database Connection – Domain level
- Default Vault – System level

Adding Complex Keys to the System Configuration Service

Important! Before adding complex keys, make sure you backup all configuration setting files in the Configuration Settings Data directory.

To add a complex key:

1. Define the complex key in an XML editor, e.g., Visual Studio. While writing the definition of a key, make sure that the XML syntax is valid and that the key name is a valid SmarTeam key name.
2. Use the editor's validation tool to ensure that the XML syntax is valid.
3. Open the SmarTeam Configuration Editor and find the relevant key.
4. If the key definition for this key does not exist, select the Add option to add it as described below.
5. If the key definition for this key already exists, append the new definition to the end of the existing definition using the Edit option as described below.
6. Copy the key definition from the XML editor, and paste it into the multi-line text box in the Configuration Editor.
7. For example, to add a new submenu containing two commands to the Application bar, add the following entries in the **<applicationBar>** key:

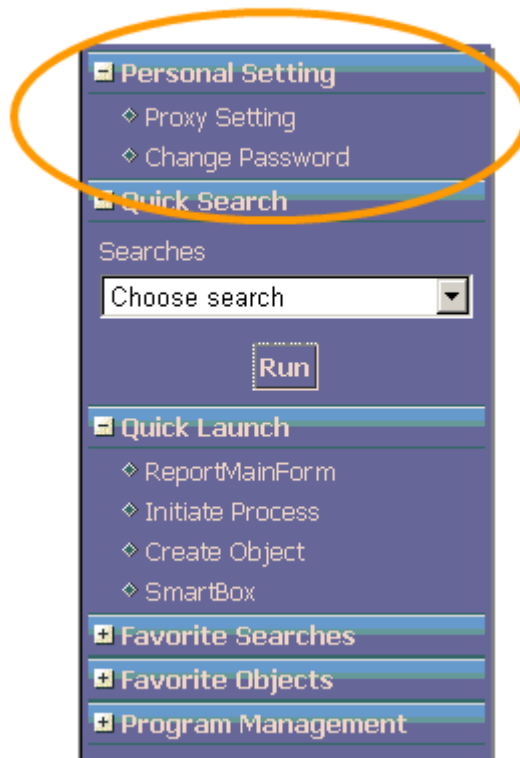
```
<menuItem>
    <commandInternalName>ud.ProxySettings</commandInternalName>
    <URL>/Views/UserDefinedTools/StatusReport.aspx</URL>
</menuItem>
<menuItem>
    <commandInternalName>ud.ChangePassword</commandInternalName>
    <URL>/Views/UserDefinedTools/StatusReport.aspx</URL>
</menuItem>
```

8. Save the changes.

Important! After adding **each** key, it is highly recommended to run the SmarTeam application and test the changes you made.

Example:

The following example shows how to add a new submenu in the SmarTeam – Web Editor application bar.



To define this new submenu in the SmarTeam Configuration Editor:

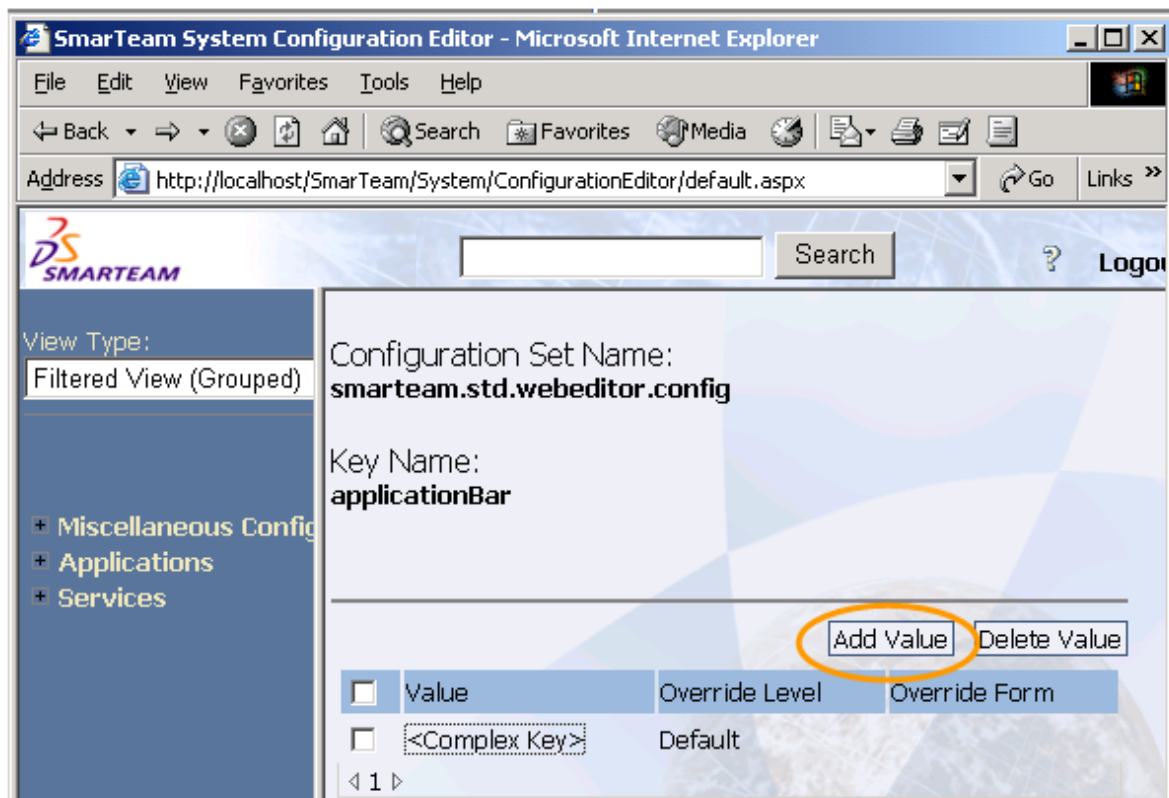
1. Open an XML editor and write the following code:

```

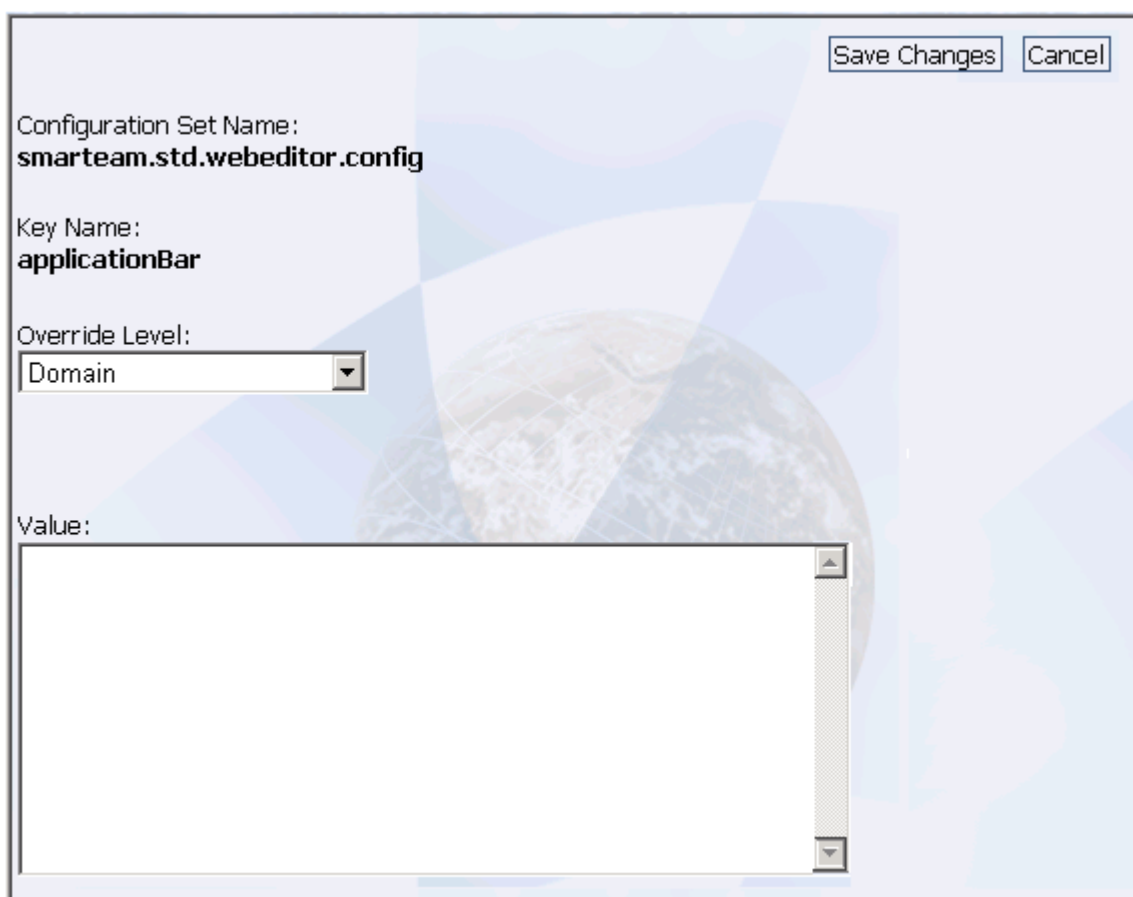
ApplicationBar.xml*
<subMenu>
  <roles>
    <role>
      <id>Default</id>
      <menuItems>
        <menuItem>
          <commandInternalName>ud.ProxySettings</commandInternalName>
          <URL>/Views/UserDefinedTools/StatusReport.aspx</URL>
        </menuItem>
        <menuItem>
          <commandInternalName>ud.ChangePassword</commandInternalName>
          <URL>/Views/UserDefinedTools/StatusReport.aspx</URL>
        </menuItem>
      </menuItems>
    </role>
  </roles>
  <subMenuId>mniPersonalSettings</subMenuId>
</subMenu>

```

2. Open SmarTeam Configuration Editor and find the key <applicationBar>.



3. Click on the “Save changes” button in the top right corner of the page.



-
4. Restart SmarTeam Services.
 5. Restart the SmarTeam – Web Editor and test your changes.

Adding a New Key to the System Configuration

To add a new key to the system configuration:

1. Create a schema file in the folder in which the current version of SmarTeam is installed: **[Smarteam HOME]\ConfigurationSettings\Schemas**
2. In this schema file, define the new or updated key. This file must contain only the changes for this version and not the complete data. For example, if the original version contained two keys and you want to add one more key in this version, the file for this version will contain the new key only and not all three keys.
3. Add a new xml file in the corresponding location under the data folder ([Smarteam HOME]\ConfigurationSettings\Data). This XML should contain delta information only (according to the schema file).

Important! In any event, it is prohibited to change SmarTeam's schema files!

Writing to the System Configuration Service using SmarTeam API (session smConfig)

In V5R10, users could add their own configuration parameters for customizing their applications. These parameters were written into the SmarTeam INI files, e.g., SMTEAM32.INI with separate headers, or created their own INI files that could be accessed using the SmConfig API functionality.

In the current version, the following rules should be applied when creating user-defined configuration parameters.

The following names are reserved names and CANNOT be used for manipulating user-defined configuration keys. Using these names will cause serious performance degradation.

smteam32.ini
SMGRID32.INI
SMGRDC32.INI
SCRGEN32.INI
SmVlt32.ini
SmWorkFlow.ini
SmERPSyncServer.ini
ServerSafeScripts.ini
WebServerSafeScripts.ini
smwiza32.ini
UpgradeSmartDatabase.ini
SmartDesk.ini

If one of the .INI files listed above is entered into the IniFileName property of SmConfig, it will turn to the System Configuration instead.

It is not recommended to manually update SmarTeam configuration parameters using SmarTeam API.

System parameters that need to be constantly updated at runtime should be stored in individual INI files located on the end-user's workstation.

Defining the Default Connection for the SmarTeam Database for Multiple Clients

The default client connection to a SmarTeam database is stored in the local computer registry and not in the System Configuration Service. This means that defining the default database at the System Configuration Server will not affect client computers using this System Configuration.

In order to avoid the need to define a default connection manually for each client computer, the relevant section of the registry on the System Configuration Server must be exported and imported at each client computer (this can be done as a part of the batch procedure).

The registry setting is located in:

HKEY_CURRENT_USER\Software\SmarTeam\SmarTeam Application\Recent Databases

Using Individual Configurations for Different Users

The situation is possible, when the admin users need to use configuration completely different from what is used for regular users. The solution is to set up additional Core Services computer and point admin client workstation on this server.

To switch client from one Configuration to another, do following:

1. Go to C:\WINDOWS\Microsoft.NET\Framework\v1.1.4322\CONFIG
2. Open the file **machine.config**
3. Locate the line (there should be 3 entries like that):
"<smarteam><sessionManagement><client><url>tcp: 127.0.0.1:5607"
4. Change the TCP/IP, for example 127.0.0.1 to the TCP/IP address of the new server.

Manually Editing System Configuration XML Files

It is not recommended to edit System Configuration XML files manually (i.e., in Notepad), as this can cause the files to be corrupt.

If you do edit System Configuration XML files manually it is highly recommended to save them in UTF8 code page (UNICODE) for availability of the system configuration in far eastern languages e.g., Japanese and Chinese.

System Configuration Service

A Microsoft Windows application, which supports the system configuration. It is installed with the SmarTeam – Foundation as part of the Core Services. It connects SmarTeam applications with the information from the Configuration schema.

Configuration Schema

The configuration schema is a collection of definitions of all possible SmarTeam System Configuration keys available from the system.

The configuration schema format is a standard W3C XSD Schema. Go to <http://www.w3.org/XML/Schema> for details.

Configuring the System Configuration Service to Work with the Windows Protocol

The System Configuration Service can be configured the system configuration to work with the Windows protocol.

In the web.config file located at <SmarTeam Home Dir>\Web\System\ConfigurationEditor the following two keys appear:

```
<authentication mode="Forms">
  <forms loginUrl="Authentication/LoginPage.aspx" name=".SMARTAUTH"/>
</authentication>

<authorization>
  <deny users="?"/>
</authorization>
```

1. Change the <authentication mode="Forms"> key to <authentication mode="Windows">
2. Comment out or delete the line starting with "<forms..."
3. Comment out or delete the "authorization" tag in its entirety including the "<deny..." and its closing tag as shown in the example below.

Example:

```
<authentication mode="Windows">
<!-- forms loginUrl="Authentication/LoginPage.aspx"
name=".SMARTAUTH"/-->
</authentication>
<!--authorization>
  <deny users="?"/>
</authorization-->
```

1. In the IIS Manager go to the Default Web Site \SmarTeam\System\ConfigurationEditor.
2. Right click and select Properties.
3. Select the Security tab and click the "Integrated Windows Authentication" check box.
4. Restart IIS.

Important! You DO NOT have to change to "Windows" authentication in the Authentication Manager.

Configuring Core Services on a Multi-Network Card Machine

The connectivity used by SmarTeam for communication between the Foundation clients and the server is **.NET Remoting**.

In order that Foundation services (Session Management and System Configuration) installed on a machine with multiple network cards work correctly, binding configurations must be applied manually.

The TCP channels in each one of Foundation services should be bound to the same network card.

In order to configure binding for the network card, you need to edit the **SessionManagementRemoting.config.xml** and **SystemConfigurationRemoting.config.xml** located in the **SmarTeam\bin** folder.

Each xml file contains tcp channel sections (beginning with <channel ref=) that contain the following code:

SessionManagementRemoting.config

```
<channel ref="tcp" port="5606">
```

```
...
```

```
...
```

```
</channel>
```

SystemConfigurationRemoting.config

```
<channel ref="tcp" port="5607">
```

```
...
```

```
...
```

```
</channel>
```

To configure binding:

1. Add the `bindTo` attribute with the IP address of the network card to the "`<channel ref="` line as follows:

SessionManagementRemoting.config

```
<channel ref="tcp" port="5606" bindTo="x.x.x.x">
```

```
...
```

```
...
```

```
</channel>
```

SystemConfigurationRemoting.config

```
<channel ref="tcp" port="5607" bindTo="x.x.x.x">
```

```
...
```

```
...
```

```
</channel>
```

Where **x.x.x.x** is the IP address of the network card.

Configuring IIS 64bit to Work with an Application running at 32bit

If you need to configure your 64 bit IIS machine to run an application at 32 bit, perform this procedure.

To configure an IIS 64bit for 32bit application:

1. Verify that ASP.NET is not installed on your server.

If ASP.NET is installed, remove it using the following example:

```
%SYSTEMROOT%\Microsoft.NET\Framework64\v2.0.50727\aspnet_regiis.exe -ua
```

2. Enter the following command to enable 32bit mode:

```
cscript %SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs  
SETW3SVC/AppPools/Enable32bitAppOnWin64 1
```

3. Install the ASP.NET.

If you uninstalled ASP.NET, you can use the following examples.

```
%SYSTEMROOT%\Microsoft.NET\Framework64\v2.0.50727\aspnet_regiis.exe -i
```

```
%SYSTEMROOT%\Microsoft.NET\Framework\v1.1.4322\aspnet_regiis.exe -i
```

4. Go to Computer Management, Services and Applications, Internet Information Services, Web service Extensions and verify that ASP.NET (32-bit) is set to: **Allowed**

XML Troubleshooting

This section contains tips to help you solve any errors that may be occurring due to problems with your XML syntax.

All XML elements must have a closing tag

In XML, all elements must have a closing tag, like this:

```
<p>This is a paragraph</p>  
<p>This is another paragraph</p>
```

XML tags are case sensitive

With XML, the tag <Letter> is different from the tag <letter>.

Opening and closing tags must therefore be written with the same case:

```
<Message>This is incorrect</message>  
<message>This is correct</message>
```

All XML elements must be properly nested

In XML all elements must be properly nested within each other like this:

```
<b><i>This text is bold and italic</i></b>
```

All XML documents must have a root element

All XML documents must contain a single tag pair to define a root element.

All other elements must be within this root element.

All elements can have sub elements (child elements). Sub elements must be correctly nested within their parent element:

```
<root>
  <child>
    <subchild>.....</subchild>
  </child>
</root>
```

Attribute values must always be quoted

With XML, it is illegal to omit quotation marks around attribute values. XML elements can have attributes in name/value pairs just like in HTML. In XML the attribute value must always be quoted as shown in the example below:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<note date="12/11/2002">
  <to>Tove</to>
  <from>Jani</from>
</note>
```

With XML, white space is preserved

With XML, the white space in a document is not truncated.

Comments in XML

The syntax for writing comments in XML is as follows:

```
<!-- This is a comment -->
```

Chapter 5: Configuring Session Management

Session Management is an integral part of SmarTeam's Core Services. In the System Configuration Service, you can define the parameters of the Session Management. The "End Time of a Session" is a parameter that defines the length of idle time of a session. After this time has passed, the Session Management automatically closes the session. This feature reduces workload on the system and subsequently releases licenses for other users. This is useful in cases where some users leave their SmarTeam applications open when not at their desks.

To configure the end-time of a System Configuration session:

1. Open the file `smarteam.std.sessionManagement.service.host.exe.config` located under the <SmarTeam Home directory>\bin directory
2. Update the value of the <expiration> tag. The value is in minutes.
3. Save the file.
4. Restart the Session Management Service.