www.wallstreetsystems.com

Wall Street Systems – **Empowering** Treasury Trade and Settlement

WALLSTREET
SYSTEMS

Wallstreet Suite

Installation and Upgrade Guide

Version 7.3.16

Information in this document is subject to change without notice and does not represent a commitment on the part of Wall Street Systems. The software and documentation, which includes information contained in any databases, described in this document is furnished under a license agreement or nondisclosure agreement and may only be used or copied in accordance with the terms of the agreement. It is against the law to copy the software or documentation except as specially allowed in the license or nondisclosure agreement. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Wall Street Systems.

Although Wall Street Systems has tested the software and reviewed the documentation, **Wall Street Systems makes herein no warranty or representation, either expressed or implied, with respect to software or documentation, its quality, performance, marketability, or fitness for a particular purpose. As a result, this software is provided "as is", and in no event will Wall Street Systems be liable for direct, indirect, special, incidental, or consequential damages from any defect in the software or by virtue of providing this documentation,** even if advised of the possibility of such damages. The documentation may contain technical inaccuracies and omissions.

The mention of an activity or instrument in this publication does not imply that all matters relating to that activity or instrument are supported by Wallstreet Suite, nor does it imply that processing of or by that activity or instrument is carried out in any particular way, even if such processing is customary in some or all parts of the industry.

The windows and screen images shown herein were obtained from prototypes during software development. The actual windows and screen images in the software may differ.

First Edition (August 2011)

This edition applies to Wallstreet Suite version 7.3.16 and to all later releases and versions until indicated in new editions or Wall Street Systems communications. Make sure you are using the latest edition for the release level of the Wall Street Systems product.

Company names, people names, and data used in examples are fictitious unless otherwise noted.

Contents

Preface

Introduction

This guide enables a qualified installation engineer to install the Wallstreet Suite software.

The engineer should be familiar with the following:

- The system design that has been agreed between the customer and Wallstreet Systems
- The prerequisite hardware and software that the system will run on
- The database that is to be prepared for and used by Wallstreet Suite
- The installation procedures described in this book.

This guide enables installation personnel to install and configure the database prior to installing Wallstreet Suite, or to perform a database upgrade to an existing installation. This guide covers all three supported databases: Oracle, MSSQL, and Sybase.

Installation personnel should have a general knowledge of the particular database and operating system platforms being used.

How to use this guide

Using the script examples

Included in this guide are scripts that you can use during installation. These are examples, and therefore use some values that you must change to suit your installation.

Copying and pasting script examples

You can copy script examples by opening this guide in Adobe Acrobat or Acrobat Reader, then pasting the clipboard contents into a plain text editor. However, if the script starts on one page and ends on another, you should ensure that you do not copy a block of text that goes over a page boundary; if you do this, you will find unwanted data being inserted into the script. To avoid this:

- In Acrobat, Select the Text tool (the mouse pointer becomes an "I" bar).
- In Acrobat, select and copy only as far as the end of the page.
- In the plain text editor, paste the text.
- In Acrobat, select and copy from the top of the next page.
- And so on.

Warning: Some code lines in this book are too long to display as a single line. For example:

```
%SrvCmd% "%fk_prefix%transd" --cmd="%fkbin%\transd.exe %stdpx%"
--dependencies="%fk_prefix%Names,%fk_prefix%mdsd"
```

When you paste this single code line into an editor, it will paste as **two** lines of code. Please remember to remove the unwanted line break.

UTF-8 encoding

From TRM version 7.1, all SQL scripts use UTF-8 encoding. When copying and pasting script examples, ensure that your text editor or terminal console supports UTF-8. Otherwise, characters that are outside the 7-bit ASCII character range can be inadvertently changed, and the database tools that you use may not warn you that a problem has occurred.

Conventions

Examples in this book are shown for both Windows and Unix operating systems. You should make the necessary changes where appropriate, for example:

- Slashes or backslashes: `/usr/bin/` for Unix, and `C:\` for Windows.
- Names of environment variables: `$ENV_NAME` for Unix, `%ENV_NAME%` for Windows.

Associated documents

Associated documents can be accessed from the Help menu of Wallstreet Suite applications.

All Wallstreet Suite user documentation can be downloaded from the Wall Street Systems Client Extranet site https://clientextranet.wallstreetsystems.com/WallstreetSuite/default.aspx.

Chapter 1 Introduction

1.1 First time installation or upgrade

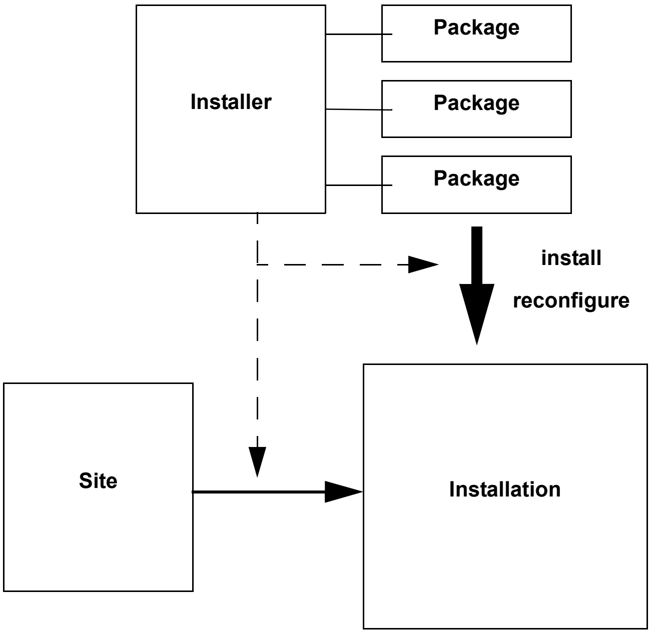
In general, the process is as follows:

1. Check hardware and software requirements.
   Chapter 2 Hardware and software requirements on page 19.
2. Download and unzip the required Wallstreet Suite package and third party support software.
   Chapter 3 Downloading Wallstreet Suite and third party software on page 25
3. **Either:** perform a first-time installation:
   a. Install a suitable database for Wallstreet Suite.
      Chapter 4 First time installation: installing the database on page 27
   b. Install Wallstreet Suite using the Suite Installer ("SIWizard")
      Chapter 5 First time installation: installing Wallstreet Suite on page 57
4. **Or:** upgrade an existing Wallstreet Suite installation:
   a. Upgrading from 7.1.0.1 onwards: use Suite Installer ("SIWizard") to upgrade the installation.
      Chapter 8 Upgrading: Wallstreet Suite on page 89
   b. Upgrading from pre-7.1.0.1, manually upgrade the existing Wallstreet Suite database.
      Chapter 9 Upgrading: database upgrade to Wallstreet Suite version 7.1.0.1 on page 101

1.2 Installation overview

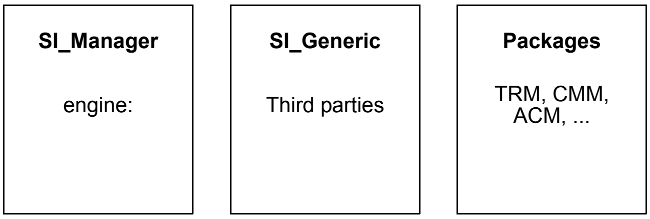
1.2.1 Suite Installer or "SIWizard"

Each Wallstreet Suite installation package contains the Suite Installer ("SIWizard") which enables you to easily install, update, and upgrade WSS products, while also providing site management.



The WSS Suite Installer package contains:

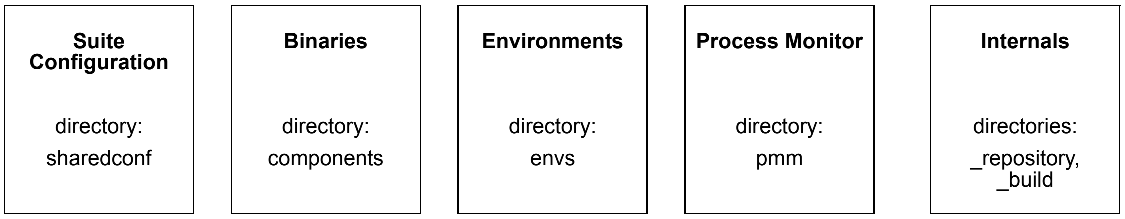
- The installer engine (available in the `SI_Manager` directory)
- Third party packages (available in the `SI_Generic` directory)
- All the WSS Suite products to be deployed by the Suite Installer (packages).



Note: Upgrading: The version of WSS Suite Installer is specific to the version of the WSS package.

1.2.2 Installation process

The installation is created and managed by the SI_Manager engine.



An installation consists of five parts:

1. **Global WSS Suite configuration:** managed through a set of property files (`sharedconf` directory)

   On customer installations, this configuration should be copied and managed only through the `site` structure.

   Configuration can be environment-specific (example: database connection details).

2. **Product binaries:** deployed at installation time. This consists of all binaries (WSS products and third party) shared across the environments.
3. **Environments:** deployed at installation and reconfiguration time.

   One environment is specific to one database and distinct from others (no port conflicts).

4. **Process Monitor:** provides management of all environment processes within one monitoring console.

   The Process Monitor is deployed in the parent directory of the installation directory.

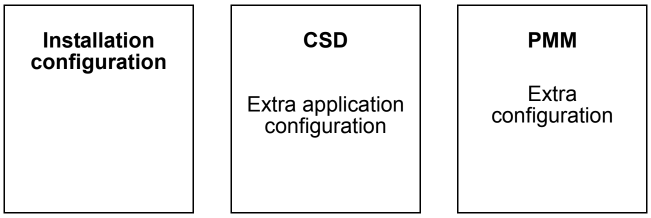
5. **Suite Installer internals:** these are directories with a "_" prefix: `_repository`, `_build` where the suite Installer stores internal data. These directories should not be modified under any circumstances.

Note: Upgrading: Any installation is relative to one specific version of WSS Suite. When upgrading, a whole new installation must be created.

The new installation contains all database upgrade scripts to be run on the customer database to upgrade.

1.2.3 Site structure

Note: This does not apply to WebSuite, which has its own system, see *7.3.4 WebSuite and resource files* on page 87.



The site structure is **mandatory** for a Wallstreet Suite installation. The site structure:

- Is a safe place which is never edited or modified by the Suite Installer.
- Contains all WSS Suite customer-specific configurations:
  - Global WSS Suite customized configuration. Common or environment specific.
  - Extra product configuration: report layouts, environment files (`pl`, `bat`), and so on.
  - Extra PMM configuration: additional servers to be managed via PMM and customized Suite servers.

The site structure is never directly accessed by the Suite Installer: it is copied into the installation using the Suite Installer.

The site directory structure must basically follow the installation structure.

Note: Upgrading: site upgrade must performed manually.

Chapter 2 Hardware and software requirements

This chapter gives guidelines for minimum hardware requirements, and software requirements for the latest version of Wallstreet Suite. It also gives an overview of the Suite Installer architecture and main concepts, and tells you how to download the Wallstreet Suite package and install the third party software required by Wallstreet Suite.

2.1 Guidelines for minimum hardware requirements

2.1.1 CPU and memory

Configuration	CPU for database	CPU for WSS processes	Memory for database	Memory for WSS processes	Total CPU count	Total memory
Small	2	2	4G	5G	4	9G
Medium	4	4	6G	12G	8	18G
Large	8	8	8G	18G	16	24G

Client software installation

- A Citrix server with 8GB of RAM can handle up to 12 users.
- Power users of Wallstreet Suite reporting or the Treasury Monitor application may need dedicated machines rather than being clustered.

General guidelines

Where possible, it is better to use multiple small computer units rather than a single large one.

Minimum configuration for starting Wallstreet Suite Installer

Wallstreet Suite Installer is a Java-based application, which requires about 2GB of memory for some operations, such as database upgrades.

This is why Wallstreet Suite Installer has minimum memory requirements: 2048MB under Unix and 1536MB under Windows).

If the workstation does not have enough available memory, Wallstreet Suite Installer crashes with following error:

```
Error occurred during initialization of VM

Could not reserve enough space for object heap

Could not create the Java virtual machine.
```

As a workaround, a lower value can be put inside the `WSSInstaller` script for the Java `-Xmx` parameter. This should enable startup of the Wallstreet Suite Installer so that you can at least install the software. See *5.4.4 Launching Suite Installer* on page 60.

Note: Database upgrade operations require a large amount of memory, and should be run from a server or workstation having the required amount of memory.

Application server:

The application server and associated interfaces require:

- 1.5GB RAM for small to medium installations.
- 12GB of RAM for large installations dealing with large volumes.

On large installations, a 64bit Unix or 64bit Windows should be used, together with the 64bit WSS package.

2.1.2 Disk space

The following figures are guidelines for estimating disk sizing. They do not include disk space taken by the operating system.

Server side Suite installation

- WSS Software : 8GB (including 2,5GB disk used by package and installer).
- WSS logs : depending on the verbosity level required, this can take several GB.
- Third party software (RDBMS client, JDK): 5GB.

Database server

- Database software RDBMS client: 5GB
- Datafiles: minimum 4GB for an empty database increasing to 150GB for a medium-sized Wallstreet Suite database
- Database backups: same disk space required as for the database.

Windows client side

- Third party software (RDBMS client, JDK): 5GB
- WSS Client Software : 4.5GB (including 2.5GB disk used by WSS package and installer).

2.2 Software requirements

Client desktop	
Operating system	Windows 7
	Windows Vista 64/32 bit
	Windows 2003 SP2
	Windows 2008 R1 or R2
	Windows 2000 SP4
	Windows XP, SP2 or SP3
Note: The Windows client desktop build (GUIs) is available from the Wallstreet Suite packages for Windows only. The Windows client desktop build is 32bit (Windows 32bit package only) and can be run on either a 32bit or 64bit platform.	
Browser	IE7.x
	IE8.x
	Firefox 3.x

Application server		Processor	Wallstreet Suite Build
Operating system	Solaris 10	SPARC	64 bit
	HP-UX B.11.23U ia64	ia64	64 bit (see note)
	Windows Server 2003, SP2	x86	32 bit
	Windows Server 2003, SP2	x86_64	64 bit (see note)
	Windows Server 2008 R1, R2	x86	32 bit
	Windows Server 2008 R1, R2	x86_64	64 bit (see note)
	RedHat Enterprise Linux 4.x 64 bit	x86_64	64 bit
	RedHat Enterprise Linux 5.x 64 bit	x86_64	64 bit
	Note: Certain optional components of the Suite may not work on this platform, owing to third party limitations. Please contact WSS Product Management to get more specific information.		
Server	Oracle Weblogic Server 10g (10.3)		
	IBM WebSphere AS 7		
	Tomcat 6.x		
Java Development Kit	JDK 6		

Databases	
Oracle	Oracle 10g to 10gR2
	Oracle 11g (11gR2 from WSS version 7.3.10)

Databases	
Microsoft SQL Server	MSSQL 2005, SP3
	MSSQL 2008 SP1 R1, R2
	Note: Cash Management (CMM): JSQL Connect licenses must be purchased by the end customer.
Sybase ASE	Sybase 12.5.4
	Sybase 15.0
	Note: Cash Management (CMM) is not supported.

2.2.1 Recommended platforms

The client operating system

Can be any combination of:

- Windows Server 2003 SP2 / Windows Server 2008 R1/R2 (Citrix / Windows Terminal Services)
- Windows XP, SP2 or SP3

For customers with heavy web traffic, the application servers can be fronted with one of these:

- IIS 6.x, IIS 7.x
- Apache 2.x

RDBMS platforms

- Oracle 11g on any Tier 1 Oracle OS (e.g. Windows, Solaris, RedHat Enterprise Linux, HP-UX)
- MSSQL 2008 R1 SP1 / R2 on any Windows Server platform.

Application server stacks

- Solaris 10, Glassfish, Oracle 11g Client
- Windows 2003 /2008, Glassfish, Oracle 11g Client
- Windows 2003/2008, Glassfish, MSSQL 2008 Client
- Red Hat Enterprise Linux 5.x 64bit, Glassfish, Oracle 11g Client
- Solaris 10, Oracle Weblogic Server 10, Oracle 11g Client
- Windows 2003/2008, Oracle Weblogic Server 10g/11g, Oracle 11g Client
- Windows 2003/2008, Oracle Weblogic Server 10g/11g, MSSQL 2008 Client
- Red Hat Enterprise Linux 5.x 64bit, Oracle Weblogic 10g/11g, Oracle 11g Client

2.3 Minimum memory requirements at Wallstreet Suite startup

	WSS server processes: physical size in MB at startup										
wss package	**SDM**	**ACM**	**Web Suite**	**Web Real Time**	**Dash board**	**TRM swift**	**DMM**	**TRM**	**Swift Connectivity**	**Suite Commons**	**Total WSS**
Windows 32b	96	146	299	126	64	394	38	1,035	55	121	2,374
Windows 64b	121	765	729	159	172	864	39	705	205	478	4,239
Solaris 64b	267	603	470	597	174	720	60	962	113	197	4,161
Linux 64b	31	557	879	202	190	977	57	917	205	557	4,573
HP-UX 64b	334	563	880	150	106	645	41	1,004	199	1,360	5,281

Important: These memory usage values are the *minimum* required at startup, and may increase dramatically when the system handles large data volumes. Note that these values do not include swap memory.

Chapter 3 Downloading Wallstreet Suite and third party software

3.1 Downloading the Wallstreet Suite package

Follow these steps:

1. Log onto the WSS Customer Support site.
   Either
   WRS: `http://customerservices.trema.com/service_enu/`
   Or
   WSS Download Center (still in test phase): `https://downloads.eu.wallstreetsystems.com/`
2. On the Update Information Service page, follow the links to the current version of the Wallstreet Suite software.
3. Locate the WSS_Installer directory, and download the package that corresponds to the operating system of your Application Server:
   - WallStreet Suite Itanium Package
   - WallStreet Suite Solaris Package
   - WallStreet Suite Linux Package
   - WallStreet Suite Windows 32b Package
   - WallStreet Suite Windows 64b Package

3.2 Installing the necessary third party software

The following third party software is required before installing the Wallstreet Suite.

- **RDBMS Client Libraries: Oracle and Sybase**

  The client libraries for your selected RDBMS need to be installed on both the Wallstreet Suite server and any Wallstreet Suite clients. For Oracle, the client-side software package is available from the vendor; for Sybase, the client-side packages are supplied by WSS on a separate CD-ROM.

- **Java Development Kit JDK 1.6**

  A Java Development Kit is required to run Wallstreet Suite. The operating system and the application server type determine which JDK is required: Sun, IBM or HP:

Operating System	Application server		
	Tomcat or Weblogic	Websphere	None (but Java required for Suite Installer)
Windows	SUN JDK `jdk1.6.0_`*`nn`*	Not supported	SUN JDK `jdk1.6.0_11`
Linux RHEL x86_64	SUN JDK `jdk1.6.0_`*`nn`*	IBM JDK	SUN JDK `jdk1.6.0_11`
Sun Solaris Sparc	SUN JDK `jdk1.6.0_`*`nn`*	IBM JDK	SUN JDK `jdk1.6.0_11`
HP HP-UX Itanium	HP JDK `java6`	Not supported	HP JDK `java6`

A 64bit JDK should be used, except where a Wallstreet Suite 32bit Windows package is to be deployed.

Important: We recommend JDK 1.6.0.14 (this is the Wall Street Systems officially certified version). A newer update of JDK 1.6 might cause errors.

SUN	`http://java.sun.com`
IBM	`www.ibm.com/developerworks/java/jdk`
HP	`www.hp.com/go/java`

Note: The path to the JDK should not include any spaces or points (full stops); when installing, you should not accept the default path `Program Files` under Windows, nor any points in subdirectory names.

An external Java(TM) Cryptography Extension is used by Suite Installer, so ensure that your JDK contains this. It is available from:
`https://cds.sun.com/is-bin/INTERSHOP.enfinity/WFS/CDS-CDS_Developer-Site/en_US/-/USD/ViewProductDetail-Start?ProductRef=jce_policy-6-oth-JPR@CDS-CDS_Developer`

Make note of where you installed the JDK (on the configuration sheet). You use this later to set the JDK home path in the Wizard-style version of Suite Installer, or in the `ts.urlpath.software.jdk.dir` property for the command-line version of Suite Installer.

- **Microsoft .NET Framework 3.5**

  This is only required if you run the Process Monitor GUI application (`pm_admin.exe`) GUI application, as opposed to the command line version (`pm_cmd.exe`). See *Chapter 6 Managing Wallstreet Suite processes* on page 73.

- **Weblogic 10.3 (optional)**

  Wallstreet Suite web application servers can be deployed over Tomcat or Weblogic. Tomcat is provided in the package, Weblogic is not. If you decide to use Weblogic, you must first install BEA Weblogic Server 10:

  a. Install the Weblogic server suite in `c:\bea` on Windows, or `/usr/local/bea` on Solaris.

  b. Make a note of where you installed Weblogic (on the configuration sheet).

  You will use it later to set the `ts.urlpath.software.weblogic.dir` property.

Note: After completing the installation using the BEA Weblogic Wizard, do **not** manually create a domain/admin server. This is handled by Suite Installer.

- **WebSphere 7.0 (optional)**

Note: Websphere products must be installed as root user in a Unix/Linux operating system.

Install WebSphere specifying required username and password. These credentials are used for Websphere Admin console and are not related to the Websuite application.

Important: The connection between the HTTP Web server and the Websphere server must be validated. For this purpose, the 'hello' test page has to be displayed successfully through Internet Explorer. Try URL pattern: http://<web server host>/hello. If this page access fails, please fix the IBM Web Server plug-in configuration first before proceeding.

Chapter 4 First time installation: installing the database

4.1 Introduction

Note the following considerations and limitations for the database:

- The database used with Wallstreet Suite is a customer asset. As such, putting its configuration, operation, maintenance, and administration under the control of a qualified DBA is essential. It is the DBA's responsibility to ensure that the implementation of the database used with Wallstreet Suite reflects customer policies and procedures.
- The amount of memory to dedicate to the database instance is not directly related to the database size. However, take into account the largest possible transaction by considering the biggest table within the database. Allocate approximately 100 MB of memory to the operating system since the machine is dedicated to database usage.
- It is recommended that you use a disk array on which I/O balancing is handled transparently.

Now go to the relevant section:

4.2 Oracle database installation on page 27

4.3 MSSQL database installation on page 37

4.4 Sybase database installation on page 43.

4.2 Oracle database installation

4.2.1 Before installing the database

Before installing the database, create directories for data and log files, for example `/disk1/oradata` and `/disk2/oradata`. For performance reasons, you should place the directories on different physical disks unless a disk controller or disk array is present.

4.2.2 Installing Oracle

Follow the procedures in the Oracle documentation to install the Oracle client and server components on the computer(s) designated in your system environment as the Wallstreet Suite Database Server. Follow these steps:

1. Modify certain specific specific UNIX kernel parameters: see the Oracle documentation for your specific UNIX platform.
2. Create the UNIX group (for example `oracle`) and user (for example `dbo`) for Oracle.
3. Follow the Oracle recommendations for Oracle Flexible Architecture (OFA), and structure your database disks using these guidelines:
   - Oracle binary files
     `ORACLE_HOME=$ORACLE_BASE/product/`*`n.n.n`*
   - Oracle admin files
     `$ORACLE_BASE/admin/<$ORACLE_SID>/admin files`

- Oracle data files (to be spread across the available disks): see *Tablespaces* on page 29. Install the Oracle database software and the prerequisite patch releases.

4. Install the Oracle database software and

- For version 10g, ensure that it is at version 10gR2.
- For version 11g, ensure that it is at version 11gR1.

4.2.3 Oracle database installation

4.2.3.1 Creating the database instance

You can create the Oracle database instance either:

- With the Oracle DBCA utility (recommended): use the Wallstreet Suite template `ts_oracle_10g_windows.dbt` (Windows) or `ts_oracle_10g_unix.dbt` (Unix) for version 10g, or `ts_oracle_11g_windows.dbt` (Windows) or `ts_oracle_11g_unix.dbt` (Unix) for version 11g. These files can be found in this directory of your Wallstreet Suite installation unzipped package:

  ```
  SI_Generic/db-build-framework_<version>-r<build-number>_<platform>.<zip|tar.gz>/
  oracle/dbca
  ```

Note: When using the Oracle DBCA utility to create the database instance, you may see the following error messages:

```
ORA-22973: size of object identifier exceeds maximum size allowed
ORA-29807: specified operator does not exist
ORA-04043: object USER_LOCK does not exist
```

These errors can be safely ignored, so click **Ignore** in the dialog where each message appears.

After using this utility, proceed to *4.2.3.2 Configuring the database instance* on page 30.

Or:

- Manually: follow the steps below.

4.2.3.1.1 Manual instance creation

Parameters to be set

In the event that the Oracle instance will be created manually, or via scripts, the following parameters must be set to the required values, and minimum defined requirements must be met.

Note: These parameters must be set exactly as shown below.

```
characterSet=AL32UTF8
nationalCharacterSet=AL16UTF16


Init Parameters:
  nls_length_semantics=CHAR
  undo_management=AUTO
  undo_suppress_errors=FALSE
```

Note: These parameters must use values greater than or equal to the values shown below.

```
optimizer_features_enable=9.2.0
db_block_size=8K
large_pool_size=16M
query_rewrite_integrity=TRUSTED
query_rewrite_enabled=TRUE
undo_retention=1800
processes=200
open_cursors=800
session_cached_cursors=300
sga_max_size=256M
sort_area_size=1048576
sort_area_retained_size=524288
db_keep_cache_size=16M
db_recycle_cache_size=16M
log_buffer=1048576
utl_file_dir=/tmp
```

Tablespaces

All the tablespaces listed below must be created, or must already exist. It is recommended that you enable autoextend for the tablespaces SYSTEM, USERS, INDX, and WORK.

Note: The relevant Wallstreet Suite template file in the `SI_Generic/db-build-framework_<version>-r<build-number>_<platform>.<zip|tar.gz>/oracle/dbca` directory of your Wallstreet Suite unzipped package creates all the required tablespaces, as well as the USERLOCK package (see *Userlock package* on page 29).

Tablespace name	Minimum available space (MB)	
	TRM	**With ACM**
SYSTEM	1000	
TOOLS	50	
UNDOTBS01	200	
INDX	1000	+250MB per 10,000 entries
USERS	1500	+250MB per 10,000 entries
TEMP	500	
WORK	400	+200MB

The tablespace sizes can be adjusted to reflect specific customer requirements.

Userlock package

As an Oracle user with DBA privileges, load the Oracle user lock feature supplied in the Oracle installation:

```
@?/rdbms/admin/userlock.sql
```

4.2.3.2 Configuring the database instance

4.2.3.2.1 User DBO

The following two steps must be performed as an Oracle user with DBA privileges:

1. Create the user DBO. This user will own all schema objects of TRM and ACM:

   ```
   CREATE USER DBO IDENTIFIED BY trema DEFAULT TABLESPACE USERS TEMPORARY TABLESPACE
   TEMP;
   ```

Note: The choice of this username is flexible, and this user will become the owner of all the Wallstreet Suite schema objects. It is recommended that you use the name DBO.

2. Run the SQL script `${FK_HOME}/share/oracle/dbo_rights.sql` which assigns the necessary database permissions to the DBO user.

4.2.3.3 Performance tuning

Performance tuning of the instance parameters is required regularly:

- During the implementation phase; this **must** be done before making it the production database.
- On the production database, especially when large amounts of data are entered.

Note: The remaining steps in the process of building a Wallstreet Suite database are most commonly completed from the server that runs the Wallstreet Suite application.

4.2.4 Building a new database

1. TRM is configured through the use of environment variables defined in configuration files stored in the `$FK_HOME/share/environments` directory. Run the `eval` command for your system to set the environment variables in one go:

   ```
   eval `/usr/trema/fk73/bin/environ -e fkprod_ora -h /usr/trema/fk73 -f`
   ```

2. Check the connection to the Oracle database (replace `<password>` with the `dbo` user's password):

   ```
   sqlplus dbo/<password>@$FK_DB_SERVER
   ```

   If the connection fails, check that:

   – tnsnames.ora contains a proper connection specification.

   – The Oracle client is properly installed.

   – The database server you are connecting to is running.

   – There is no network error.

3. Build a new database, using the perl script `build.pl` provided.

   Read the warning below before executing the build statement that follows. Replace `<password>` with the password of the user `dbo`):

   ```
   cd $FK_HOME/share/oracle
   ./build -U dbo -P <password> -D $FK_DB_SERVER -a -v -k -w 2> buildnewdb.log
   ```

   For more information about the `build.pl` script and its options, see *Appendix D The build script* on page 139.

Warning: The **-w** option enables the database for the Static Data Manager (SDM). You should include the **-w** option **only** if you are going to install SDM.

4. Recompile the invalid DBO's stored objects:

```
sqlplus dbo/<password>@$FK_DB_SERVER
START /usr/trema/fk72/share/oracle/migration/fast_recomp.sql
START /usr/trema/fk73/share/oracle/migration/fast_recomp.sql
```

5. Ensure that there are no invalid objects for the user dbo:

```
SELECT object_type, count(object_name) QTY_INVALID FROM user_objects WHERE status
= 'INVALID' GROUP BY object_type;
```

6. Check the TRM database version:

```
SELECT value FROM dbo.Configuration WHERE name LIKE '%Version%';
```

7. Update the Configuration table to allow DBO to log in:

```
SELECT id,name,value FROM Configuration WHERE name LIKE '%login%';
```

Example response:

```
        ID NAME                           VALUE
---------- ------------------------------ -------------
       554 allow dbo login                false
```

```
UPDATE Configuration set value = 'true' WHERE id = 554;
COMMIT;
```

8. Back up your database using the Oracle `expdp` tool. It is important to set `NLS_LANG` to the correct values; to query `NLS_LANG`, see *4.2.8 Exporting and importing a TRM database* on page 35.

```
ORACLE_SID=fkprod; export ORACLE_SID

NLS_LANG=american_america.AL32UTF8; export NLS_LANG

expdp system/system12 dumpfile=fkprod_fk7nnnn.dmp schemas=dbo logfile=expdbo.log
```

4.2.5 Installing the ACM schema objects

CM is installed into the same database (and the same database schema) as TRM. Once the TRM database is installed, you simply add the ACM objects to those created for TRM.

To build the database objects, ACM uses a wrapper script named `acmbuild` around the TRM build script, and this adds some functionality to the TRM `build` script. The command line options for `acmbuild` are the same as for the TRM `build` script. For more information about the TRM `build` script and its options, see *Appendix D The build script* on page 139.

4.2.5.1 Setting up the ACM prerequisites

4.2.5.1.1 Evaluating the environment

TRM is configured through the use of environment variables defined in configuration files stored in the `$FK_HOME/share/environments` directory. Run the `eval` command for your system to set the environment variables in one go:

```
eval `/usr/trema/fk73/bin/environ -e fkprod_ora -h /usr/trema/fk73 -f`
```

4.2.5.1.2 Testing the database connection

Check the connection to the Oracle database (replace <password> with the `dbo` user's password):

Unix:

```
sqlplus dbo/<password>@$FK_DB_SERVER
```

Windows:

```
sqlplus dbo/<password>@%FK_DB_SERVER%
```

When successfully connected, type `exit` and press ENTER.

If the connection fails, check that:

- `tnsnames.ora` contains a proper connection specification.
- Oracle client is properly installed.
- The database server you are connecting to is running.
- There is no network error.

4.2.5.1.3 Changing directories to the appropriate folder

Change to the correct directory which contains the `acmbuild` script.

Unix:

```
cd $FK_HOME/share/oracle
```

Windows:

```
cd /d %FK_HOME%\share\oracle
```

4.2.5.2 Loading the ACM schema objects

4.2.5.2.1 Executing the build script

Build ACM database objects using the Perl script `acmbuild.pl` provided:

Unix:

```
./acmbuild -D $FK_DB_SERVER -U dbo -P <password> -y

$FK_HOME/share/oracle/objects.inc.acm -t \* -V \* -p \* -g \* -u -v
```

Windows:

```
perl acmbuild.pl -D %FK_DB_SERVER% -U dbo -P <password> -y

%FK_HOME%\share\oracle\objects.inc.acm -t * -V * -p * -g * -u -v
```

For more information about the `build.pl` script and its options, see *Appendix D The build script* on page 139.

4.2.5.2.2 Recompiling invalid stored objects

Start a SQL session and execute the following command to recompile DBO’s invalid stored objects:

Unix:

```
sqlplus dbo/<password>@$FK_DB_SERVER
```

Windows:

```
sqlplus dbo/<password>@%FK_DB_SERVER%
```

Unix:

```
START $FK_HOME/share/oracle/migration/fast_recomp.sql
```

Windows:

```
START %FK_HOME%\share\oracle\migration\fast_recomp.sql
```

4.2.5.2.3 Checking for invalid objects

Ensure that there are no invalid objects for the user `dbo`:

Unix:

```
sqlplus dbo/<password>@$FK_DB_SERVER
```

Windows:

```
sqlplus dbo/<password>@%FK_DB_SERVER%
```

```
SELECT object_type, count(object_name) QTY_INVALID FROM user_objects WHERE status

= 'INVALID' GROUP BY object_type;
```

4.2.5.3 Completing the ACM schema object installation process

4.2.5.3.1 Checking the version of the ACM schema

Ensure that the correct version of the ACM schema was installed. To check the current version of the ACM schema, execute the following command:

```
select max(SPKGVersion) from ACMVersion where SParameters = 'finished';
```

Notes

- It is necessary to escape the asterisks (*) with the backslashes for non-Windows platforms.
- Replace <password> with the `dbo` user password in the command line examples.
- Owing to a bug in Oracle database server version 9.2.0.6 and some later versions, views using ANSI joins become invalid when compiled under a user other than the view's owner. If you use a recompilation script run under the SYS user, then ACM views become invalid. Recompile ACM views under the DBO user only. For the bug reference, see https://metalink.oracle.com/metalink/plsql/showdoc?db=Bug&id=3466980
- Despite the fact that ACM Application Server uses the `ORACLE_SID` environment variable to populate the `ACM_JDBC_URL` environment variable, the `sqlplus` command must be run with a database name according to the local `tnsnames.ora` configuration file. Typically `ORACLE_SID` contains the Oracle database `SID` while the `FK_DB_SERVER` environment variable contains a TNS alias from the local `tnsnames.ora`.

4.2.6 Installing the CMM schema objects

CMM is installed into the same database (and the same database schema) as TRM. Once the TRM database is installed, you simply add the CMM objects to those created for TRM.

To build the database objects, CMM uses a wrapper script named `cmmbuild` around the TRM build script, and this adds some functionality to the TRM `build` script. The command line options for `cmmbuild` are the same as for the TRM `build` script. For more information about the TRM `build` script and its options, see *Appendix F The cmmbuild.pl tool (pre version 7.3.0)* on page 149.

4.2.6.1 Setting up the CMM prerequisites

4.2.6.1.1 Evaluating the environment

TRM is configured through the use of environment variables defined in configuration files stored in the `$FK_HOME/share/environments` directory. Run the `eval` command for your system to set the environment variables in one go:

```
eval `/usr/trema/fk73/bin/environ -e fkprod_ora -h /usr/trema/fk73 -f`
```

4.2.6.1.2 Testing the database connection

Check the connection to the Oracle database (replace <password> with the `dbo` user's password):

Unix:

```
sqlplus dbo/<password>@$FK_DB_SERVER
```

Windows:

```
sqlplus dbo/<password>@%FK_DB_SERVER%
```

When successfully connected, type `exit` and press ENTER.

If the connection fails, check that:

- `tnsnames.ora` contains a proper connection specification.
- Oracle client is properly installed.
- The database server you are connecting to is running.
- There is no network error.

4.2.6.1.3 Changing directories to the appropriate folder

Change to the correct directory which contains the `cmmbuild` script.

Unix:

```
cd $FK_HOME/share/oracle
```

Windows:

```
cd /d %FK_HOME%\share\oracle
```

4.2.6.2 Loading the CMM schema objects

4.2.6.2.1 Executing the build script

Build CMM database objects using the Perl script `cmmbuild.pl` provided:

Unix:

```
./cmmbuild -D $FK_DB_SERVER -U dbo -P <password> -y

$FK_HOME/share/oracle/objects.inc.acm -t \* -V \* -p \* -g \* -u -v
```

Windows:

```
perl cmmbuild.pl -D %FK_DB_SERVER% -U dbo -P <password> -y

%FK_HOME%\share\oracle\objects.inc.acm -t * -V * -p * -g * -u -v
```

For more information about the `build.pl` script and its options, see *Appendix D The build script* on page 139.

4.2.6.2.2 Recompiling invalid stored objects

Start a SQL session and execute the following command to recompile DBO’s invalid stored objects:

Unix:

```
sqlplus dbo/<password>@$FK_DB_SERVER
```

Windows:

```
sqlplus dbo/<password>@%FK_DB_SERVER%
```

Unix:

```
START $FK_HOME/share/oracle/migration/fast_recomp.sql
```

Windows:

```
START %FK_HOME%\share\oracle\migration\fast_recomp.sql
```

4.2.6.2.3 Checking for invalid objects

Ensure that there are no invalid objects for the user `dbo`:

Unix:

```
sqlplus dbo/<password>@$FK_DB_SERVER
```

Windows:

```
sqlplus dbo/<password>@%FK_DB_SERVER%
```

```
SELECT object_type, count(object_name) QTY_INVALID FROM user_objects WHERE status

= 'INVALID' GROUP BY object_type;
```

4.2.6.3 Completing the CMM schema object installation process

4.2.6.3.1 Checking the version of the CMM schema

Ensure that the correct version of the CMM schema was installed. To check the current version of the CMM schema, execute the following command:

```
select max(SPKGVersion) from CMMVersion where SParameters = 'finished';
```

Notes

- It is necessary to escape the asterisks (*) with the backslashes for non-Windows platforms.
- Replace <password> with the `dbo` user password in the command line examples.
- Owing to a bug in Oracle database server version 9.2.0.6 and some later versions, views using ANSI joins become invalid when compiled under a user other than the view’s owner. If you use a recompilation script run under the SYS user, then ACM views become invalid. Recompile ACM views under the DBO user only. For the bug reference, see https://metalink.oracle.com/metalink/plsql/showdoc?db=Bug&id=3466980
- Despite the fact that ACMApplication Server uses the `ORACLE_SID` environment variable to populate the `ACM_JDBC_URL` environment variable, the `sqlplus` command must be run with a database name according to the local `tnsnames.ora` configuration file. Typically `ORACLE_SID` contains the Oracle database `SID` while the `FK_DB_SERVER` environment variable contains a TNS alias from the local `tnsnames.ora`.

4.2.7 Housekeeping and maintenance

4.2.7.1 Backups

The first thing to decide when creating a backup/recovery strategy is how to handle the Archive Log files, as any production database is running in ARCHIVELOG mode. Normally the customer organization already has a good backup/recovery strategy for its systems, so it is more a matter of adding TRM databases in to the existing plan.

4.2.8 Exporting and importing a TRM database

Back up and restore a TRM database by exporting and importing it using the Oracle `exp` and `imp` utilities. Using these utilities is a useful way to copy a production database to create a test database.

When importing or exporting data, it is important that you use the correct values for the NLS_LANG environment variable. To find out the database language settings, run this query:

```
set heading off
set pagesize 0
set verify off
set feedback off
set termout off
set echo off
set embedded off
set recsep off
SELECT a.value ||'_'|| b.value ||'.'|| c.value FROM NLS_DATABASE_PARAMETERS a,
NLS_DATABASE_PARAMETERS b, NLS_DATABASE_PARAMETERS c
WHERE a.parameter = 'NLS_LANGUAGE'
AND   b.parameter = 'NLS_TERRITORY'
AND   c.parameter = 'NLS_CHARACTERSET';
```

4.2.8.0.1 Exporting

Replace the example values for `NLS_LANG` below with the values that correspond to your database as obtained by the query you ran in *4.2.8 Exporting and importing a TRM database* on page 35.

```
ORACLE_SID=fk_prod; export ORACLE_SID
NLS_LANG=american_america.AL32UTF8; export NLS_LANG
cd /oracle/dumps
exp system/manager FULL=Y DIRECT=Y RECORDLENGTH=65535
FILE=fk_prod_full_20May05.dmp LOG=impdbo.log
expdp system/system12 dumpfile=fkprod_fk7nnnn.dmp schemas=dbo logfile=expdbo.log
```

4.2.8.0.2 Importing

Replace the example values for `NLS_LANG` below with the values that correspond to your database as obtained by the query you ran in *4.2.8 Exporting and importing a TRM database* on page 35.

```
ORACLE_SID=fk_test; export ORACLE_SID
NLS_LANG=american_america.AL32UTF8; export NLS_LANG
cd /oracle/dumps
imp system/manager FROMUSER=dbo TOUSER=dbo
FILE=fk_prod_full_20May05.dmp LOG=impdbo.log
expdp system/system12 dumpfile=fkprod_fk7nnnn.dmp schemas=dbo logfile=expdbo.log
```

Whenever you import data into the TRM database, you **must** run the `after_load_db.sql` script afterwards:

```
sqlplus -s dbo/<password> << EOS
@/usr/trema/fk65fk72/share/oracle/migration/after_load_db.sql
@/usr/trema/fk65fk73/share/oracle/migration/after_load_db.sql
```

This script initializes Oracle sequences to their proper values, recompiles invalid objects, computes statistics for TRM tables, and enables triggers that could have been disabled during import.

4.2.8.1 Rebuilding of indexes and stored procedures

Warning: If you do not have SDM installed, do **NOT** include the **`-w`** option in any of the following `build` statements.

TRM comes with a "nightly" script that can be used to rebuild database objects, typically stored procedures and indexes, when needed. This can be done using a shell script that sets the environment variables and then executes the following command:

```
[acm|cmm]build.pl -U <login> -P <password> -D <database> -i all -p all -v -w >
rebuild.log
```

Check the log after this rebuild to verify that objects were correctly rebuilt.

If you change any TRM database object, you should revalidate all stored procedures using `fast_recomp.sql`. The example below can be run from another machine, from the `fkadmin` account (replace `<password>` below with the password for the user `dbo`):

```
#!/bin/sh
ORACLE_HOME=/opt/oracle/10.2.0.1.0; export ORACLE_HOME
ORACLE_SID=fkprod_database ; export ORACLE_SID
sqlplus -s dbo/<password> << EOS
```

```
@/usr/trema/fk65fk73/share/oracle/migration/fast_recomp.sql
exit
EOS
```

4.2.8.2 DBVerify

TRM databases should be checked using the Oracle provided DBVerify tool, called dbv, on a daily basis, usually before the backup is run. The output file from DBVerify should be checked and verified so that possible datafile corruptions are found as soon as possible.

The tool can be run against online datafiles or backups.

To list the datafiles:

```
sqlplus dbo/<password>
```

```
SELECT file_name FROM dba_data_files;
dbv FILE=/oradata2/fk65fk72/users01.dbf BLOCKSIZE=8192
dbv FILE=/oradata2/fk65fk73/users01.dbf BLOCKSIZE=8192
```

The following sample shell script runs DBVerify against all datafiles in a database:

```
ORACLE_HOME=/opt/oracle/9.2.0.6;export ORACLE_HOME
ORACLE_HOME=/opt/oracle/10.2.0.1.0;export ORACLE_HOME
ORACLE_SID=fkprod_database;export ORACLE_SID
LOG_FILE=/tmp/dba_export.log; export LOG_FILE
if [ -f ${LOG_FILE} ]; then rm ${LOG_FILE}; fi
echo "[`date`] BEGIN Checking datafiles integrity."
DATAFILE=`sqlplus -s "/ as sysdba" << EOS
set heading off
set pagesize 0
set verify off
set feedback off
set termout off
set echo off
set embedded off
set recsep off
SELECT file_name FROM dba_data_files;
exit
EOS`
export DATAFILE
for df in $DATAFILE
do
echo "DBVerify: Checking datafile $df ....." \ >>
${LOG_FILE}
dbv file=$df blocksize=8192 logfile=/tmp/dbv.log
echo `grep Corrupt /tmp/dbv.log` \ >> ${LOG_FILE}
done
echo "[`date`] END of Datafiles integrity check."
# more ${LOG_FILE}
```

4.3 MSSQL database installation

4.3.1 Before installing the database

Before installing the database, do the following:

- In preparation for installation, create directories for data and log files, for example `X:\MSSQL\data and Y:\MSSQL\log`. For performance reasons, you should place the directories on different physical disks unless a disk controller or disk array is present.

- The installation program will prompt you to select the Services Account and Authentication Mode for use by the database. See your system administrator or database administrator (DBA) for the correct values to select.

4.3.2 Database considerations and limitations

Note the following considerations and limitations for the database:

- The database used with Wallstreet Suite is a customer asset. As such, putting its configuration, operation, maintenance, and administration under the control of a qualified DBA is essential. It is the DBA’s responsibility to ensure that the implementation of the database used with Wallstreet Suite reflects customer policies and procedures.
- The amount of memory to dedicate to the database instance is not directly related to the database size. However, take into account the largest possible transaction by considering the biggest table within the database. Allocate approximately 100 MB of memory to the operating system since the machine is dedicated to database usage.
- It is recommended that you use a disk array on which I/O balancing is handled transparently.

4.3.3 Installing the Server

You can install either Microsoft SQL Server 2005, 2008, or 2008 R2.

4.3.3.1 Microsoft SQL Server 2005 or 2008

Follow the installation procedure described in the documentation that comes with the Microsoft SQL Server software. During the installation procedure, do the following:

1. Select `Local Computer` as the computer name.
2. When prompted for the type oif install, select `Create a new instance of SQL Server, or install Client Tools`.
3. In the **Installation Definition** dialog, select `Server and Client Tools`.
4. In the **Instance Name** dialog, click `Default`.
5. In the **Setup Type** dialog, select `Custom`.
6. In the **Select Components** dialog, select everything except `Code Samples`.
7. In the **Services Accounts** dialog, select either `System account` or a specific account for SQL Server. What you select here though depends on how your administration is set up.
8. In the **Authentication Mode** dialog, the default to select is `Mixed Mode`. What you select here depends on how your administration is set up.
9. Enter the password and confirm it for the `sa` login.
10. in the **Collation Settings** dialog, select `SQL Collations`, and highlight

    `Dictionary order, case-sensitive, for use with 1252 character Set`

    Or in the case of 2008: `SQL_Latin1_CP1_CS_AS`.

Important: The Collation Settings are very important: once confirmed, you can change them only by reinstalling the server. Ensure that you select case sensitivity.

11. In the **Network libraries** dialog, enter and select the following:
    - Select `Named Pipes: Named Pipe name: \\. \pipe\sql\query`
    - Select `TCP/IP Sockets: Port number: 1433`

- Select `Multi-Protocol`

Note: If you have another SQL server on your machine, check that the TCP/IP port numbers are not the same.

4.3.3.2 Creating a new database manually

You can use the Suite Installer to create a new empty database and ensure backwards compatibility with SQL Server 2000. But if you want to do this manually then:

1. Use Microsoft SQL Server Management Studio to create a new database:
   - Select the Database directory in Microsoft SQL Server Management Studio, then right-click it to bring up a menu, then select **New Database...**.
   - Create and name the database (for example `WSStest01`) and log files.
2. Backwards compatibility with SQL Server 2000 is ensured by activating the command:

   ```
   sp_dbcmptlevel <db_name>, 80
   ```

4.3.4 Building the database

This is handled by the Suite Installer.

If the database owner has to connect to the database via TRM, then a specific change must be made to a database table.

Using Microsoft Query Analyzer, execute this command:

```
update Configuration set value='true' where id=554
```

4.3.5 Housekeeping and maintenance

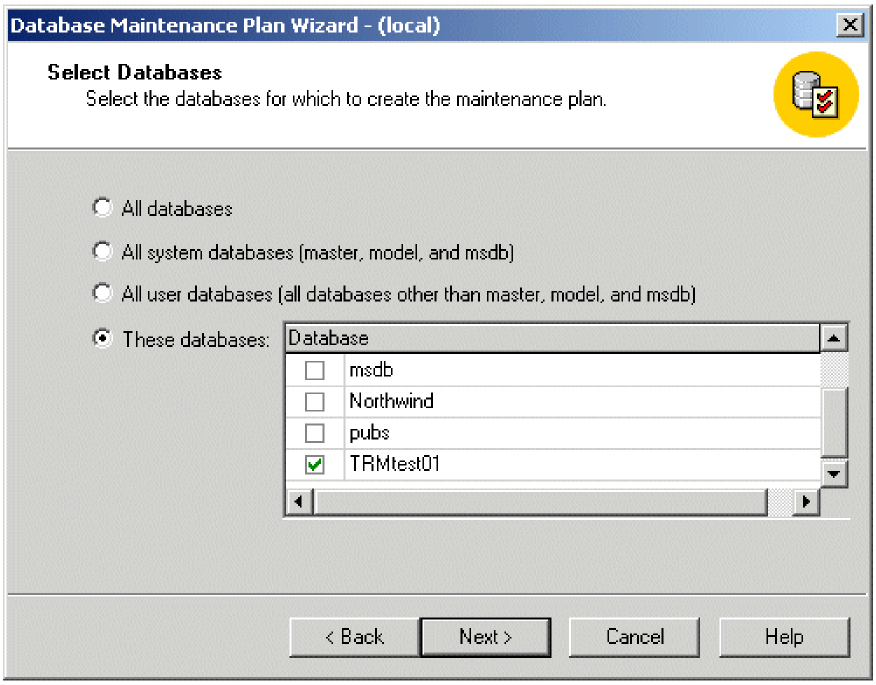
4.3.5.1 Defining a database maintenance plan

The database administrator must implement a database maintenance plan to perform these common administrative tasks:

- Reorganize data and index pages (recommended task).
- Remove unused space from database files (optional task).
- Check database integrity (recommended task).

- Backup the database (mandatory task).

1. In Microsoft SQL Server Management Studio, right click on your database, then select **All Tasks - Maintenance Plan...**
2. Select the TRM database only, then click **Next**..



Database Maintenance Plan Wizard - (local)

Select Databases

Select the databases for which to create the maintenance plan.

○ All databases

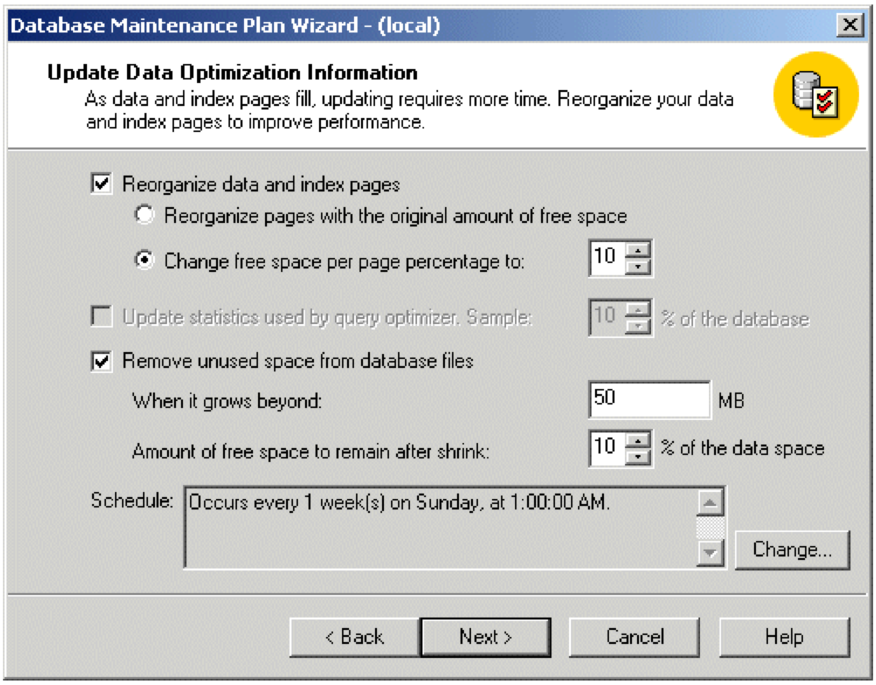
○ All system databases (master, model, and msdb)

○ All user databases (all databases other than master, model, and msdb)

● These databases:

	Database
□	msdb
□	Northwind
□	pubs
☑	TRMtest01

3. In the following panels, configure the values as required. The screenshots that follow are given as examples, and show suggested values; the database administrator must choose the most suitable parameters and schedules.



Database Maintenance Plan Wizard - (local)

Update Data Optimization Information

As data and index pages fill, updating requires more time. Reorganize your data and index pages to improve performance.

☑ Reorganize data and index pages

○ Reorganize pages with the original amount of free space

● Change free space per page percentage to: 10

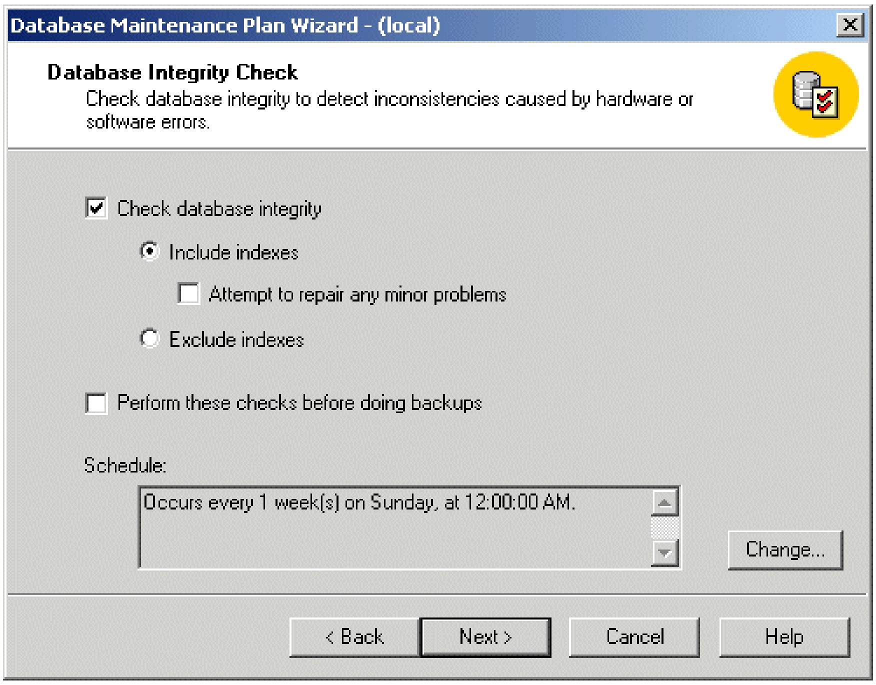
□ Update statistics used by query optimizer. Sample: 10 % of the database

☑ Remove unused space from database files

When it grows beyond: 50 MB

Amount of free space to remain after shrink: 10 % of the data space

Schedule: Occurs every 1 week(s) on Sunday, at 1:00:00 AM.



Database Integrity Check

Check database integrity to detect inconsistencies caused by hardware or software errors.

☑ Check database integrity

● Include indexes

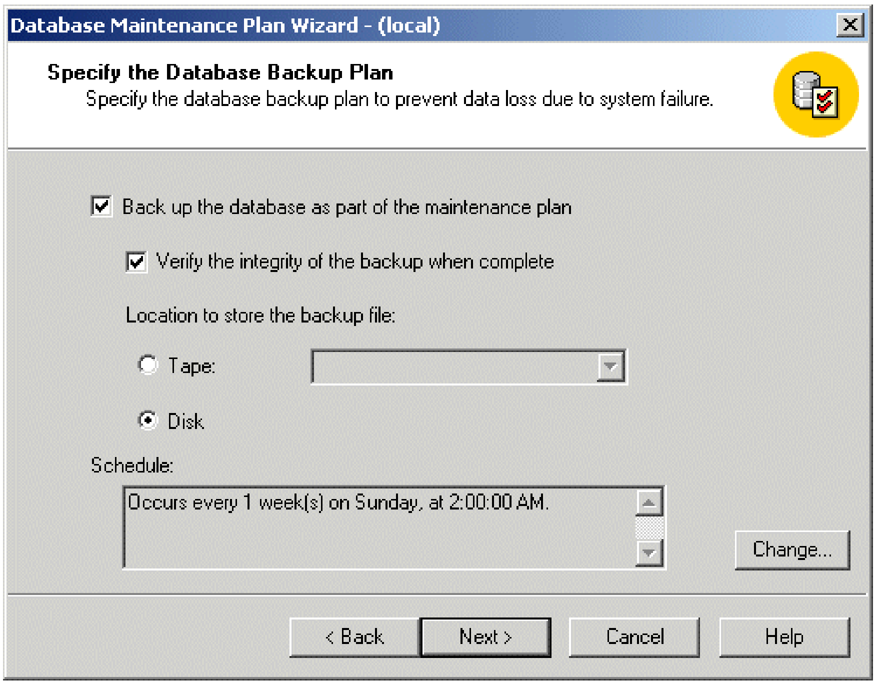
□ Attempt to repair any minor problems

○ Exclude indexes

□ Perform these checks before doing backups

Schedule:

Occurs every 1 week(s) on Sunday, at 12:00:00 AM.



Specify the Database Backup Plan

Specify the database backup plan to prevent data loss due to system failure.

☑ Back up the database as part of the maintenance plan

☑ Verify the integrity of the backup when complete

Location to store the backup file:

○ Tape:

● Disk

Schedule:

Occurs every 1 week(s) on Sunday, at 2:00:00 AM.

4.3.5.2 Rebuilding indices and stored procedures

Warning: If you do not have SDM installed, do **NOT** include the `-w` option in any of the following `build` statements.

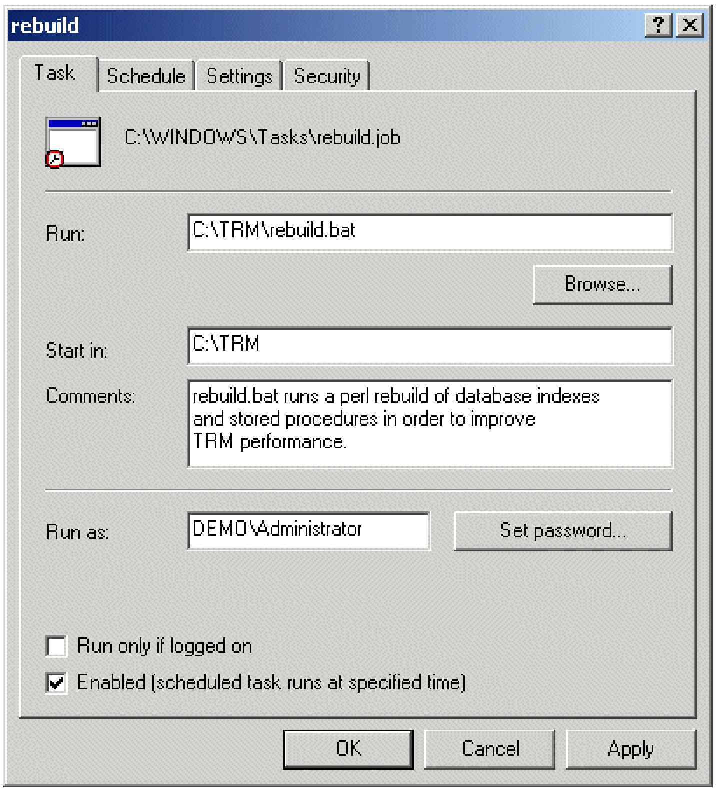
TRM comes with a script that can be used to rebuild database objects, typically stored procedures and indexes, on a regular basis. This can be done using the Windows Server scheduler, with a `.bat` command file that sets the environment variables and then executes the following command:

1. Copy and paste the line below to a file called `rebuild.bat` and save it in your release directory (for example `C:\TRM`)

```
call %WSS_HOME%\bin\trm-<ENV>-shell.bat
cd %FK_HOME%\share\mssql
perl build.pl -U login -P password -D database -i all -p all -v -w >
%FK_HOME%\log\rebuild.log 2>&1
```

2. Schedule the task by selecting **Start - Settings - Control Panel - Scheduled Tasks - Add Scheduled Task**, which runs the scheduled tasks wizard.

3. Choose appropriate parameters for schedule options. Once sheduled, your task appears in the scheduled tasks and can be modified by right-clicking its icon and selecting **Properties**:



C:\WINDOWS\Tasks\rebuild.job

Run: C:\TRM\rebuild.bat

Start in: C:\TRM

Comments: rebuild.bat runs a perl rebuild of database indexes and stored procedures in order to improve TRM performance.

Run as: DEMO\Administrator

□ Run only if logged on

☑ Enabled (scheduled task runs at specified time)

4. Check the `%FK_HOME%\log\rebuild.log` after this rebuild to verify that objects were correctly rebuilt.

4.4 Sybase database installation

Follow the procedures in this section to install Sybase.

Note: If your license of Sybase was not obtained through Wall Street Systems, you need to know the Sybase Customer Authentication String (CAS).

4.4.1 Checking the shared memory

Databases benefit from very large shared caches of data in some cases. The `/etc/system` file needs to be edited to support shared memory. If you have 1024 MB of RAM in a computer serving only as a Sybase server, leave 64 MB for the operating system and allocate the remaining 960 MB as shared memory. The value for 192 MB shared memory is calculated as 960*1024*1024=1006632960.

Add the following line to `/etc/system` if you want 960 MB of shared memory:

```
set shmsys:shminfo_shmmax=1006632960
```

Reboot the computer for the change to take effect.

4.4.2 Setting up Sybase group, user, and home account

To make sure that Sybase product files and directories are created with consistent ownership and privileges, one UNIX user (typically the **Sybase system administrator** with read, write, and execute privileges) should perform all installation, configuration, and upgrade tasks. To create a Sybase system administrator UNIX account, choose an existing account or create a new account and assign it a user ID*,* group ID, and password. This account is sometimes called the **Sybase user** account.

Use the UNIX `admintool` database manager to create:

- A Sybase group with `gid=30`
- A Sybase user with `uid=30`
- A home directory where the Sybase software will be installed. Select, for example, `/opt/sybase` as the Sybase home directory.

The shell must be the Bourne shell.

4.4.3 Defining the Sybase environment

Add the following lines to the `.profile` of the Sybase user:

```
.SYBASE.sh

DSQUERY=SERVERNAME;export DSQUERY

SYBASE="/opt/sybase/SQL/current";export SYBASE

SYBASE_OCS=<see note below>

PATH=/bin:/usr/sbin:$SYBASE/$SYBASE_OCS/bin:.;export PATH
```

Note: `SYBASE_OCS=OCS-12_5 for Sybase version 12.5.4.`
`SYBASE_OCS=OCS-15_0` for Sybase version 15.0.3.

You can use either the `$SYBASE/SYBASE.sh` shell or the `$SYBASE/SYBASE.csh` shell.

4.4.4 Organizing disks

Discuss disk organization with Trema technical support. Issues that have an impact on the disk layout are redundancy and performance.

One recommended approach is to use a RAID module with several disks in order to increase performance. It also eliminates the need to put the index segments on separate disks, since the index segments will be distributed over several disks anyway. An extra "hot spare" disk will also have a positive impact on recovery issues.

The solution given below is an example of one way of organizing your disks.

4.4.4.1 Disk organization

Sybase supports file system files as well as raw devices, such as database devices.To keep the database consistent with file system files, set the `dsync` flag on database devices. Mirroring can be carried out with the Sybase built-in mirroring function or with some other disk mirroring software, for example SUN's On-line Disk Suite or Veritas Volume Manager.

Disks should be mirrored using Raid 0. This provides absolute duplication of the data on the mirror device and is fault tolerant. (Media failure causes a switch over to the mirror device.) It is not recommended to use Raid 5 devices for Sybase.

The partitions need to be at least the following sizes for an average installation:

```
sybase_master   200MB
sybase_sysprocs 200MB
sybase_data     4GB for each database
sybase_log      2GB for each database
```

The `sybase_data` (database) and `sybase_log` (log of database) slices should be on separate disks for recovery reasons. If you do not use separate disks, there is no point in having separate partitions.

Warning: Do not use the first cylinder of a disk for raw data partitions. If anybody changes the disk label on the disk, the first cylinder will be overwritten.

4.4.5 Creating links to Sybase devices

You need to create links from `$SYBASE/dev/` to the raw devices you are planning to use. This will enable you to get a clear view of which Sybase devices you have. You will also be able to restore the Sybase master and user databases to disks which have hardware addresses that are different from the disks the backups were taken from.

4.4.5.1 Sybase device links

Create the following links (depending on your setup):

```
mkdir /opt/sybase/dev
cd /opt/sybase/dev
ln -s /dev/md/rdsk/d4 master
ln -s /dev/md/rdsk/d5 sysproc
ln -s /dev/md/rdsk/d6 data1
ln -s /dev/md/rdsk/d7 log1
```

As the UNIX superuser, `root`, change the ownership of the partitions you are planning to use to the Sybase user:

```
chown sybase /opt/sybase/dev/*
```

4.4.6 Creating Sybase directory structure and current Link

When creating the directory structure under the Sybase home directory (for example `opt/sybase`), you need to create a `./SQL/` directory where the software is installed under its version number. A link named `current` must point to the current version of the software. These actions must be carried out by the Sybase user.

To install Sybase, do the following:

```
mkdir -p /opt/sybase/SQL/sybase12
cd /opt/sybase/SQL
ln -s sybase12 current
```

The purpose of this is to make it easy to find the software on all systems. The `isql` command should, for example, always be accessible under:

```
/opt/sybase/SQL/current/$SYBASE_OCS/bin/isql
```

4.4.7 Loading the Sybase software

Untar the Sybase tar.gz file to a temporary directory, then run the `setup` graphical interface (requires X-Windows).

4.4.8 Applying the latest Emergency Bug Fixes (EBFs)

EBFs for Sybase can be downloaded from http://support.sybase.com/ if you have a Sybase support agreement directly with Sybase. If your Sybase support is supplied by Trema, the EBFs are provided by Trema.

The following steps take you through the installation of an EBF.

Note: Check the cover letter that comes with the EBF to be sure that this is the correct method for your setup.

1. If necessary, back up your databases.
2. Shut down your ASE server (Sybase dataserver) as follows:

   ```
   isql -Usa -P""
   1> shutdown
   2> go
   ```

3. Back up the `$SYBASE` directory (if needed).
4. Read the cover letter related to this EBF.
5. Use gnu `unzip` to unzip the file:

   ```
   gunzip EBF9999.tgz
   ```

6. Then untar the file:

   ```
   tar xvf EBF9999.tar
   ```

7. Run the setup script (graphical mode) to install the EBF on your Sybase intallation.
8. After loading the contents of the ESD Rollup, boot the ASE:

   ```
   cd install
   ./startserver -f ./RUN_SERVERNAME
   ```

9. (For an existing ASE server only) Run the `installmaster` script:

   ```
   isql -U sa - P <sa password> -S <server name> -n -i
   $SYBASE/$SYBASE_ASE/scripts/installmaster -o <output file>
   ```

10. Check that you are running the right version:

```
isql -Usa -P""
1> select @@version
2> go
```

ASE is now running the updated Sybase software.

4.4.9 Building the Sybase ASE (srvbuild)

To build up the Sybase ASE (also known as the ASE, Adaptive Server Enterprise), use `srvbuild`:

- For Sybase version 12.5.4: `$SYBASE/ASE-12_5/bin/srvbuild`
- For Sybase version 15: `$SYBASE/ASE-15_0/bin/srvbuild`

Follow these steps:

1. Choose the Adaptive Server (usually SYBASE), and the Backup Server (usually SYBASE_back).
2. Change the page size to 8KB.
3. Specify the master device: `/opt/sybase/dev/master`
4. Check that the size of the device is the size of the disk partition.
5. Create a 30 MB master database.
6. Specify the `sysprocs` device: `/opt/sybase/dev/sysproc`
7. Check that the size of the device and database is the size of the disk partition.
8. Define a port number (2024) in the interfaces file.
9. Select OK.
10. Configure the Backup Server.
11. Set port number to 2025.
12. Build the servers.
13. Choose character sets and sort order:

    Character set: Unicode 3.1 UTF8

    Sort order: Binary sort order for the ISO 1064 6-1, UTF8 multibyte encoding character set (UTF8).

The Sybase system administrator ID (user ‘sa’) is automatically created when Sybase is installed.

4.4.10 Initializing devices

A database device stores the objects that make up the database. The term *device* does not necessarily refer to a distinct physical device; it can refer to any piece of a disk (such as a disk partition), or a file in the file system that is used to store the database and its objects. Each database device or file must be prepared and made known to the Sybase ASE before it can be used for database storage. This process is called initialization and is divided into two steps.

4.4.10.1 Step 1: Calculating the size of the devices (prtvtoc)

The size of the data devices can be calculated by dividing the sector count by 4. The sector count is available with the `prtvtoc` command.

Running this command:

```
prtvtoc /opt/sybase/dev/data1
```

produces the following output:

```
* /opt/sybase/dev/data1 partition map
*
* Dimensions:
* 512 bytes/sector
* 80 sectors/track
* 19 tracks/cylinder
* 1520 sectors/cylinder
* 3500 cylinders
* 2733 accessible cylinders
*
* Flags:
* 1: unmountable
* 10: read-only
*
*                         First     Sector     Last
* Partition Tag Flags     Sector     Count     Sector
       2     5   00         0      4154160    4154159
       3     0   00         0        10640      10639
       4     0   00     10640       205200     215839
       5     0   00    215840      1843760    2059599
       6     0   00   2059600      2094560    4154159
```

The size of slice 5 in 2 KB blocks will be 1843760/4 = 460940.

4.4.10.2 Step 2: Executing the command

The next step is to initialize the Sybase data devices. Log in to Sybase with `isql` and perform the following commands (the size specified depends on your system):

```
isql -Usa -P""
1> disk init name="data1",physname="/opt/sybase/dev/data1",
2> vdevno=4,size=460940
3> go
```

4.4.11 Creating and configuring the databases

Use only half of the available disk space for the TRM database. This will leave enough room for a test database. This space can also be used for upgrading to a new version of TRM when required. The following example creates a 2200MB database on the `data1` device with a 600MB transaction log on the `log1` device (`log`). The * in `trm*` indicates the version of TRM that you are installing, for example `trm71`.

```
isql -Usa -P""
1> create database trm* on data1 = "2200M" log on log1 = "600M"
2> go
```

4.4.11.1 Configuration of the database

If you are going to use a template or Trema’s “Best Practice” database, make sure that you create your database correctly; it must be created the same way the template or Best Practice database was created. Otherwise, the log and data segments will not be correct.

4.4.12 Expanding tempdb

The ASE has a temporary database, **tempdb**. This database is used for temporary tables and any other intermediate storage needs, for example, intermediate results of SQL “group by” and “order by” commands. The space in tempdb is shared among all users of all databases on the server.

By default, the system allocated tempdb on a master device of 2 MB. You must allocate tempdb on a new device of 1024 MB (minimum) and deallocate it from the master device as follows:

- tempdb allocation:

```
disk init
         name        = "tempdb_dev",
         physname    = "/sybase/devs/tempdb.dev",
         vdevno      = 4,
         size        = 2048000 -- expressed in 2k pages, example of 4000 MB of size
         go
         alter database tempdb
         on  tempdb_dev = 1000
         go
         use tempdb
         go
```

It is important to remove tempdb allocation from the master device used for the master database (dev_master):

```
sp_dropsegment "default", tempdb, dev_master
         go
         sp_dropsegment "logsegment", tempdb, dev_master
         go
         sp_dropsegment "system", tempdb, dev_master
         go
```

4.4.12.1 Increasing the size of tempdb

To create a 1024 MB file, do the following:

```
mkdir /opt/sybase/dev
isql -Usa -P
1> disk init name="tempdbfile",
2> physname="/opt/sybase/dev/tempdbfile",
3> vdevno=5,size=512000
4> go
```

Use the stored procedure `sp_helpdevice` to check that the device is created. Then alter the database:

```
1> alter database tempdb on tempdbfile = 1024
2> go
```

4.4.13 Database segments

If you have stripped disks and do not use a separate index device, you do not need to read this section.

Segments are logical groups of disk devices (or portions of devices) where database objects can be placed. Database segments are created in the database, and these can be put on a particular database disk device. Using segments gives the system administrator or database owner increased control over the placement, size and space usage of database objects, since individual objects can be placed in specific segments. Segments can also increase ASE performance. It is usual to have a separate log segment for recovery purposes.

Consider a database that spans two disk devices and has two segments. All tables with non-clustered indexes can be created on one segment and the indexes for those tables on another. This gives an increase in performance since the reading and writing of the data and index pages can take place concurrently on the two separate physical devices.

If you spread your database over several disks, you can increase the overall performance by assigning one disk each for indexes, logs, and data. You assign disks with the `sp_addsegment` command. More details about segments and using the `sp_addsegment` command are given in Default segments on page 30 XXX.

To find out how the database is segmented, use the `sp_helpdb, sp_helpsegment` command, for example:

```
1> use trm*
2> go
1> sp_helpdb trm*
2> go
```

where trm* is the name of the TRM database, for example trm71.

4.4.14 Database options

You can control the following aspects of database behavior by using the database options available with the stored procedure sp_dboption:

- Behavior of transactions
- Defaults for table columns
- User access restrictions
- Performance of recovery and bcp (Sybase bulk copy) operations
- Log behavior.

System administrators and database owners can use these database options to configure the settings for an entire database. In comparison, note that sp_configure parameters affect the entire server and set options affect only the current session or stored procedure.

Set the following options for the TRM database:

```
1> use master
2> go
1> exec sp_dboption trm*, "select into", "true"
2> exec sp_dboption trm*, "trunc", "true"
3> go
1> use trm*
2> go
1> checkpoint
2> go
```

where trm* is theTRM version.

Note: If possible, turn off truncate log on checkpoint and dump the transactions to a dump device. This provides greater security.

4.4.15 Sybase ASE configuration parameters

The Sybase ASE configuration parameters are user-definable settings that control the behavior of this server.

Configuration parameter	Minimum value	Comment
Recovery interval in minutes	1	
Cache size	262144.0K	
Number of open databases	30	
Number of open objects	40000	
Number of open indexes	15000	
Partition groups	2048	
Number of open partitions	10000	New parameter in ASE 15.0
Number of devices	20	
Enable unicode conversions	1	
Default network packet size	1024	
Max network packet size	8192	
Number of remote connections	100	

Configuration parameter	Minimum value	Comment
Number of remote logins	100	
Number of worker processes	20	Sybase recommends not using parallel processing iwhen upgrading to ASE 15, so set this value to 0 to begin with.
Max parallel degree	12	Sybase recommends not using parallel processing iwhen upgrading to ASE 15, so set this value to 1 to begin with.
Max memory	1024000	The memory value is in 2 KB blocks
Additional network memory	256000	
Allocate max shared memory	1	
Max online engines	2	
Number of engines at startup	2	
Procedure cache size	128000	Multiply your ASE 12.5 procedure cache size by 3 for ASE 15.
Identity burning set factor	1	
CPU grace time	1500	
Number of user connections	300	
Number of locks	200000	
Deadlock checking period	1000	
Print deadlock information	1	
Optimization goal	1, allrows_oltp	Use Transactional dedicated mode instead of allrows_mix which is the default setting.
Enable literal autoparam	1	
Statement cache size	25600	

The ASE supplies default values for the configuration parameters are based on the following assumptions:

- At least 15 MB of RAM is dedicated to the ASE
- The ASE is subject to a large volume of update activity
- A small number of stored procedures are used frequently.

Because these assumptions do not apply to TRM, you must reset some configuration parameters to optimize server performance. Parameters for memory allocation, user connections, and the number of locks must be reset.

The size of the procedure cache, the number of user connections, and the amount of memory of the dataserver (Sybase ASE process) must be increased.

For Sybase version 15 only, there is a new configuration parameter:

```
1> use master
2> go
1> exec sp_configure " number of open partitions ", 100000
3> go
```

4.4.15.1 Memory use

The memory value is in 2 KB blocks and cannot exceed the value specified for shared memory in the system specification file `/etc/system`. The procedure cache value is given in 2KB blocks. For example, if you have 960 MB total memory and want a 32 MB procedure cache, execute the following commands:

```
1> use master
2> go
1> exec sp_configure "max memory", 491520
2> exec sp_configure "procedure cache", 16000
3> exec sp_configure "number of open objects", 40000
4> exec sp_configure "number of open indexes", 40000
5> go
```

4.4.15.2 Lock Manager

```
1> exec sp_configure " number of locks ", 200000
2> go
```

4.4.15.3 Buffer data cache

Extend the cache size that can be used by your ASE:

```
1> exec sp_cacheconfig ’default data cache’, ’256M’, mixed
2> go
```

4.4.15.4 User connections

The number of user connections usually needs to be increased. To increase the number to more than 60, ensure that the Sybase dataserver has been started after the command `ulimit -Sn 'ulimit -Hn'` has been executed.

For 50 user connections:

```
1> exec sp_configure "user connections", 50
2> go
```

4.4.15.5 Sybase server packet size

The Sybase server should be configured to accept a bigger packet size. This is because the TRM client normally has a large packet size pre-configured.

```
1> sp_configure "max network packet size", 8192
2> go
```

Configuring the packet size to be 8192 improves response time according to standard Trema benchmarking. For example, client reporting can be significantly faster with a packet size of 8192 compared to the default packet size of 512.

Set the environment variable SYBASE_PACKET_SIZE to 8192 on the client side. This needs to be done in conjunction with the server-side configuration described above. If this is not done, the packet size will be set to 512 and is independent of the value set for the "max network packet size" parameter on the server.

The client database connection will fail if the value set for the client-side variable SYBASE_PACKET_SIZE is greater than the server-side "max network packet size" variable.

4.4.15.6 Sybase ASE memory

To use this bigger packet size, the Sybase ASE has to be configured to have more network memory. Follow these steps to calculate the extra memory required:

1. Estimate the number of simultaneous users who will request large packet sizes, and multiply this number by the sizes their applications will request.
2. Multiply this by 3, since each connection needs three buffers.
3. Add 2 percent for overhead.
4. Round the value to the next highest multiple of 2048.

Then use the following command to set up the extra memory (the example below is for 50 users):

```
1> sp_configure "additional network memory",  1251328
2> go
```

Memory allocated with the “`additional network memory`” parameter in `sp_configure` is added to the memory already allocated to total memory. It does not affect other Sybase server memory uses.

Note: The amount of memory allocated using `additional network memory` plus the memory already allocated to total memory must **always** be less than the shared memory.

4.4.15.7 Using Sybase parallel functionality

If your server has several processors, refer to the Sybase documentation to make sure you are using them as effectively as possible.

4.4.15.8 Identity burning set factor

The identity burning set factor can cause problems with table columns which have automatic numbering when the Sybase server shuts down unexpectedly. Such table columns are defined as type *AutomaticID*. This factor sets the number of IDs stored in memory. You need to set the number kept in memory as low as possible:

```
1> sp_configure "identity burning set factor", 1
2> go
```

You can also use the system procedure `sp_chgattribute` that comes with Sybase to limit the gap value that may occur on tables that have an identity field following an abrupt stop of the server.

```
    Syntax = sp_chgattribute 'table_name', 'identity_gap', gap_value desired
```

For more information see http://www.sybase.com/detail?id=20113.

Note: This problem does not occur when the Sybase server is shut down and rebooted correctly.

4.4.15.9 "cpu grace time"

"cpu grace time" specifies the maximum amount of time in milliseconds that a user process can run without yielding the CPU before ASE releases it. The default value is 500, but the TRM Client applications require a higher value: 1500 is recommended.

```
1> sp_configure "cpu grace time", 1500
2> go
```

4.4.15.10 "optimization goal"

Sybase ASE default value for parameter optimization goal is mixed_mode, which is supposed to work with a transaction processing application or decision support system.

TRM will offer better performance with the optimization goal parameter to allrows_oltp (Transactional dedicated mode instead of allrows_mix which is the default settings)

```
1> sp_configure 'optimization goal', 1, allrows_oltp
2> go
```

4.4.15.11 Character sets and sort order

If you have an existing Sybase ASE installation, you may need to install Unicode support, and make it the default character set. See *9.3.1.1 Converting charset to UTF8* on page 105.

4.4.16 Traceflags

Turn on traceflag 15353 which disables "intelligent scan adjustment".

For this add -T 15353 to the start file.

This traceflag can solve potential problems with optimizer choosing P* index on Transactions when running Average Balance.

4.4.17 Setting up the automatic start of the Sybase ASE

Note: You must first install TRM before you can do this.

To make sure that the Sybase ASE is automatically restarted after the system has been rebooted, the startup script should be installed by running `setup.sybase`:

```
cd $FK_HOME/etc
./setup.sybase
```

4.4.18 Installing Sybase 15 on Linux RHEL 5 64-bit

There are several known issues with ASE on RHEL 5. This is the configuration you should use:

1. Install `libstc` compatibility

   Sybase requires `libstdc++.5` but RHEL 5 has `libstdc++.6` only, so install this rpm to fix `compat-libstdc++-33-3.2.3-61`.

2. Take care of the installer issue where keyboard input is not taken into account in the installer's GUI:

   a. `cd` to the `/usr/X11R6/lib/X11` directory (create it if necessary)

   b. Create a symbolic link:

   ```
   $ ln -s /usr/share/X11/XKeysymDB XKeysymDB
   ```

   c. `cd` to the `/usr/X11R6/lib/X11/locale` directory (create it if necessary)

   d. Make the following symbolic links:

   ```
   $ ln -s /usr/share/X11/locale/en_US.UTF-8 en_US.UTF-8

   $ ln -s /usr/share/X11/locale/locale.alias locale.alias

   $ ln -s /usr/share/X11/locale/locale.dir locale.dir
   ```

3. Edit the /etc/sysctl.conf file and add two parameters:

   ```
   kernel.exec-shield = 0

   kernel.randomize_va_space = 0
   ```

4. Avoid a potential crash during the startup of the backup server by setting this variable in the Sybase environment (in `.bash_profile`):

   ```
   export LD_POINTER_GUARD=0
   ```

4.4.19 Building a new database

1. TRM is configured through the use of environment variables defined in configuration files stored in the `$FK_HOME/share/environments` directory. Run the `eval` command for your system to set the environment variables in one go:

   ```
   eval `/usr/wss/fk/bin/environ -e fkprod_syb -h /usr/wss/fk -f`
   ```

2. Check the connection to the Sybase database (replace `<password>` with the `dbo` user's password):

   ```
   isql -U fk -P fkfkfk -S $FK_DB_SERVER -D $FK_DATABASE
   ```

3. Check the TRM database version (if one was loaded)

```
select value from Configuration where id = 10
go
```

4. Grant `sso_role` to the database owner login. This role is required to create other logins.

```
isql -U sa -P <sa password> -S $FK_DB_SERVER
1>grant role sso_role to fk
2>go
```

5. Give some specific system permissions. This is required by the Security Center module.

```
isql -U sa -P <sa password> -S $FK_DB_SERVER
1>use master
2>go
1>grant select on sysloginroles to public
2>go
1>grant select on syssrvroles to public
2>go
1>grant select on syslogins to public
2>go
```

6. Build a new database, using the perl script `build.pl` provided (replace `<password>` with the password of the user `dbo`):

   Read the warning below before executing the `build` statement that follows. Replace `<password>` with the password of the user `dbo`):

```
cd $FK_HOME/share/sybase
./build -S $FK_DB_SERVER -D $FK_DATABASE -U fk -P <password> -a -v -k -w
```

   For more information about the `build.pl` script and its options, see *Appendix D The build script* on page 139.

Static data management: If you want to use static data management (SDM), then you must use the `-w` option of the `build` script.

Warning: You should include the `-w` option **only** if you are going to use SDM.

4.4.20 Rebuilding stored procedures, views, and permissions

This operation can be performed as part of a database upgrade or as a maintenance operation. See *9.3 Sybase upgrade* on page 105.

Note: If you have already used the -w option of the `build` script, you must add it to the statement below.

```
cd $FK_HOME/share/sybase
./build -S $FK_DB_SERVER -D $FK_DATABASE -U fk -P <password> -v -g all -p all -V
all
```

4.4.21 Housekeeping and maintenance

4.4.21.1 Backups

The first thing to decide when creating a backup strategy is how to handle Transaction Logs. It is strongly recommended that you dump the transaction log, so you need to create a schedule for doing this. There is nothing special about backing up TRM databases, and normally the organization should already have good backup strategies; so it is a matter of adding TRM databases to the existing backup/recovery plan.

4.4.21.2 DBCC

The TRM databases should be checked using DBCC on a daily basis, usually before the backup is run. The output file from DBCC should also be checked and verified so that any inconsistencies are found as soon as possible.

4.4.21.3 Database objects maintenance

Warning: If you do not have SDM installed, do **NOT** include the `-w` option in any of the following `build` statements.

TRM comes with a "nightly" script that can be used to rebuild DB objects regularly, typically Stored Procedures and Indexes. This can be done using `crontab` in UNIX, together with a bat file that sets the environment variables and then executes:

```
cd $FK_HOME/share/sybase
./build -U <login> -P <password> -S <db server> -D <database> -i all -p all -v >
c:\tmp\buildlog.txt -w
```

You should check the log after this rebuild so that you do not have objects that have been dropped but not built for some reason. This could cause a disastrous situation, for example a unique index on a table.

Update Statistics on all the tables should be done on a daily basis.

Recreating indexes should be done regularly. The frequency depends on how big the daily changes to the database are: a new system that goes from 100 transactions to 200 transactions probably needs to be rebuilt every day, whereas a system that goes from 20 000 transactions to 20 100 transactions does not.

4.4.21.3.1 Examples

Set the environment variables for the system:

```
eval `/usr/wss/fk/bin/environ -e fkprod_syb -h -f`
```

Updating statistics

```
isql -U fk -P fkfkfk -S $FK_DB_SERVER -D $FK_DATABASE
exec UpdateStatistics
go
```

Re-creating all indexes

```
cd $FK_HOME/share/sybase
./build -v -i all -U fk -S $FK_DB_SERVER -D $FK_DATABASE -w
```

Rebuilding all stored procedures

```
cd $FK_HOME/share/sybase
./build -v -p all -U fk -S $FK_DB_SERVER -D $FK_DATABASE -w
```

Running the build script to re-create one procedure

```
cd $FK_HOME/share/sybase
./build -S $FK_DB_SERVER -D $FK_DATABASE -U fk -P fkfkfk -v -p ReadPortfolioCashflows
-w
```

Running the build script to create a table:

```
cd $FK_HOME/share/sybase
./build -S $FK_DB_SERVER -D $FK_DATABASE -U fk -P fkfkfk -v -t SetupMapData -o
~/cr_SetupMapData.sql -w
```

```
./build -S $FK_DB_SERVER -D $FK_DATABASE -U fk -P fkfkfk -v -t SetupMapData -w
```

Chapter 5 First time installation: installing Wallstreet Suite

5.1 Introduction

The Wallstreet Suite Installer enables you to install, configure, reconfigure, and upgrade the following:

- A server running all or selected parts of Wallstreet Suite (the WSS Application Server).

  This is the server that runs the real-time processes.

- The Wallstreet Suite database, normally running on a separate server.
- Clients.

  These are Windows machines that have all rich client GUIs installed (Windows and Java). This can be on the main server, or several other PCs.

- Client shortcuts

  Creates batch files on the target PC to run the rich client GUIs, and optionally creates Windows shortcuts.

5.2 Prerequisites

Linux/Unix

On a Unix/Linux machine, the Wallstreet Suite Installer requires X Server. If this is not available, you can edit configuration files instead.

Windows

To install on a Windows machine requires "administrator" permissions.

Java Development Kit

Java Development Kit (JDK) must be installed on the server.

Important: We recommend JDK 1.6.0.14 (this is the Wall Street Systems officially certified version). A newer update of JDK 1.6 might cause errors.

If the Wallstreet Suite package is a 64bit package, then the JDK must also be 64bit.

RDBMS client

An Oracle, Sybase, or SQL Server RDBMS client must be installed on the server.

If the Wallstreet Suite package is a 64bit package, then the RDBMS client must also be 64bit.

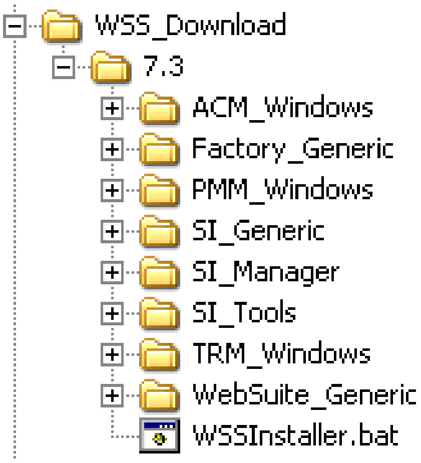
Windows clients

The rich client GUIs are available only on 32bit versions of Windows.

5.3 Expand the Wallstreet Suite package

Expand the Linux/Unix or Windows version of the Wallstreet Suite package on the WSS Application Server and on any machines that will run Wallstreet Suite (for example clients). This creates a top-level directory whose name is the version number of the software. This contains the Wallstreet Suite software and the Suite Installer software that configures the Wallstreet Suite system.

The package expands into the directory hierarchy shown in the example below. The Windows version is shown here, expanded inside a user-created directory called `WSS_download`:



```
WSS_Download
    7.3
        ACM_Windows
        Factory_Generic
        PMM_Windows
        SI_Generic
        SI_Manager
        SI_Tools
        TRM_Windows
        WebSuite_Generic
        WSSInstaller.bat
```

Important: On Unix, the Wallstreet Suite package must be extracted using **GNU** tar, as the system tar command fails on Solaris or HP-UX. As a prerequisite GNU tar must be installed on the server. The extract command must then be run as follows:

```
/usr/local/bin/tar xf ws-suite-installer_<version>-r<number>_<platform>.tar
```

5.4 Inside the Suite Installer

5.4.1 Basics and changes from previous versions

The Suite Installer is a wrapper around the existing python command used in previous versions, and enables part of the installation to be automated.

The existing commands `python installpackages/reconfigure` still exist, and can be launched from the Suite Installer.

The most important change is the removal of the `ts-create-db.bat` and `ts-upgrade-db.bat` scripts which have been replaced by generated scripts.

5.4.2 WSSInstaller options

For help on `WSSInstaller`, enter:

```
WSSInstaller.bat -help
```

This displays a list of options that you can use when launching `WSSInstaller`. The only options that are recommended for users are:

`--loglevel <1-4>`

Specifies the level of logging that occurs:

1 Error information only

2 Error and warning information

3 Info information

4 All information

`--package-file <path/package_filename>`

Enables the user to use a specified package file, for example `--package-file /<path>/suite-install-full.txt`

`--silent`

Launches the Suite Installer without the GUI. See *Automated installation* on page 61.

`--config-file`

Uses a file generated during a prevous installation to use the same parameters. See *Repeated installation* on page 61.

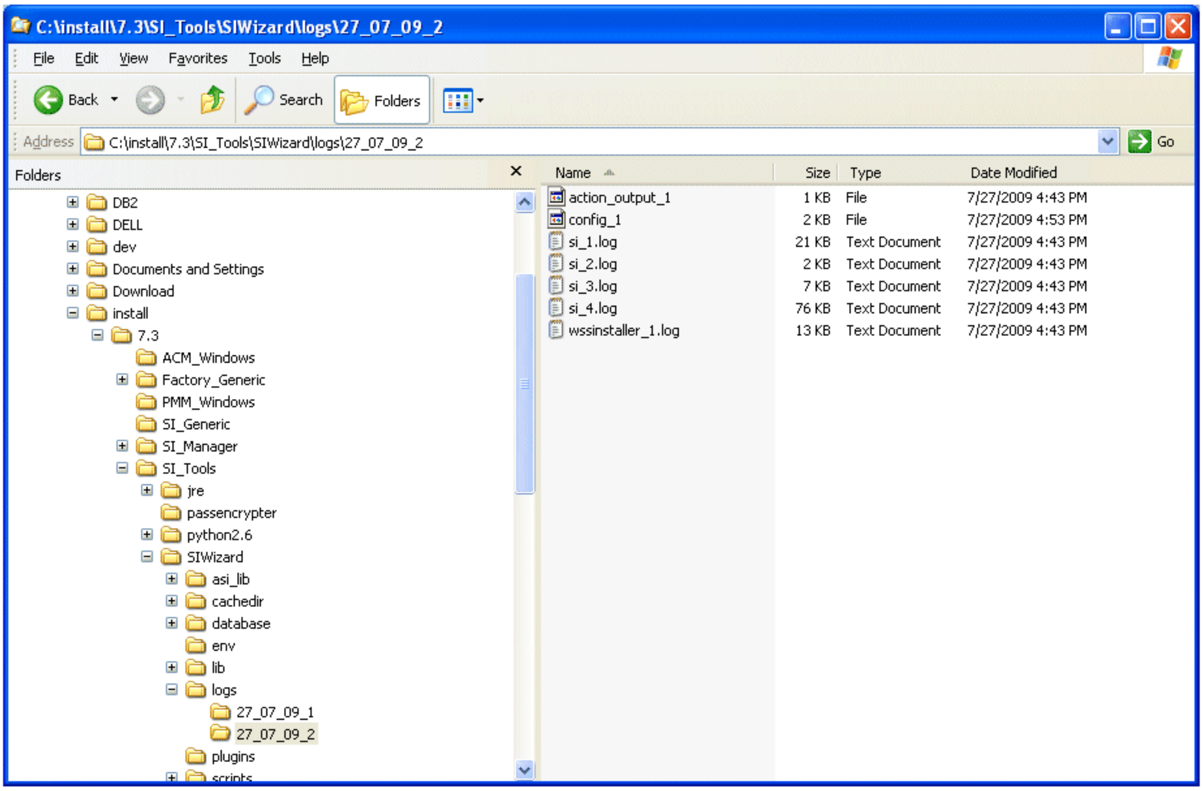
Other options

Other options that can be used for specific tasks are described in *5.4.9 Other options* on page 62.

5.4.3 Suite Installer logs

You can find the logs in the following location:

`SIWizard\logs\<date>\`



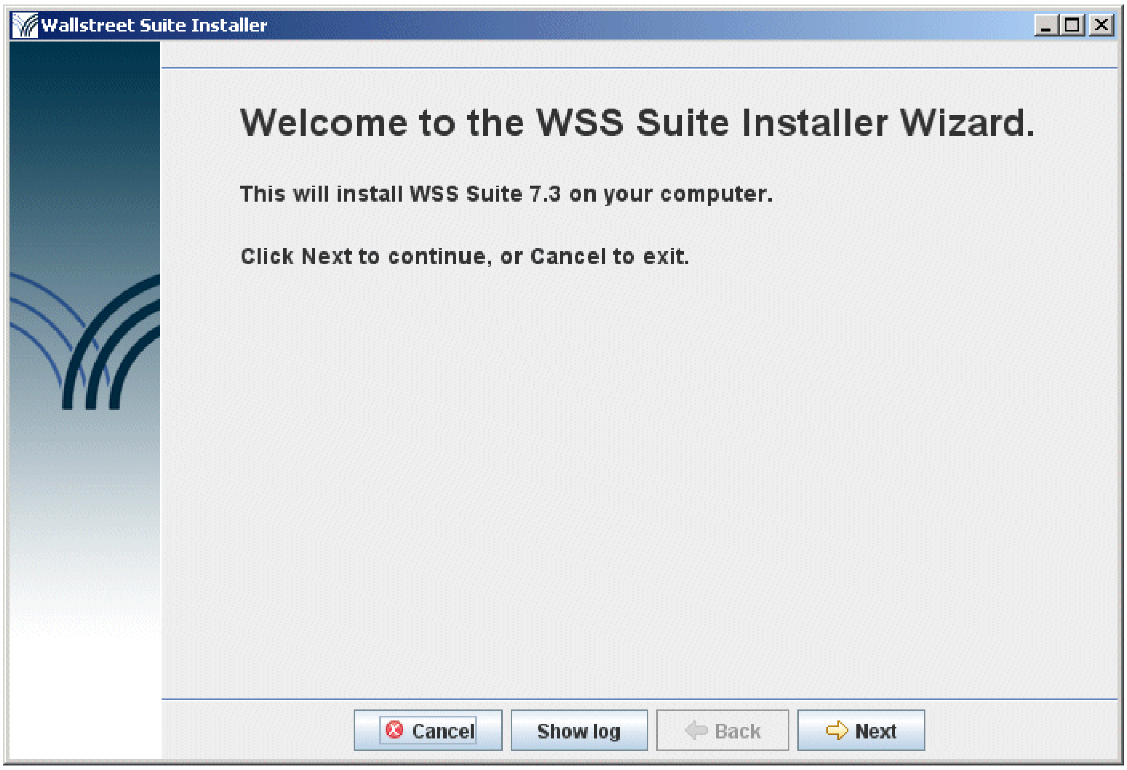
Name	Size	Type	Date Modified
action_output_1	1 KB	File	7/27/2009 4:43 PM
config_1	2 KB	File	7/27/2009 4:53 PM
si_1.log	21 KB	Text Document	7/27/2009 4:43 PM
si_2.log	2 KB	Text Document	7/27/2009 4:43 PM
si_3.log	7 KB	Text Document	7/27/2009 4:43 PM
si_4.log	76 KB	Text Document	7/27/2009 4:43 PM
wssinstaller_1.log	13 KB	Text Document	7/27/2009 4:43 PM

This `logs` directory contains all scripts and logs generated during one installation. Each launch of `WSSInstaller.sh`/`.bat` generates a new `logs` directory.

5.4.4 Launching Suite Installer

For a first-time installation, you should start by installing the WSS Application Server, then the clients.

Launch the file `WSSInstaller.bat`, or `WSSInstaller.sh`.
After a few seconds you see this:



Welcome to the WSS Suite Installer Wizard.

This will install WSS Suite 7.3 on your computer.

Click Next to continue, or Cancel to exit.

Placing the cursor over most fields, buttons, and other controls causes popup help to appear.

If you have problems starting Wallstreet Suite Installer, see *Minimum configuration for starting Wallstreet Suite Installer* on page 19.

5.4.5 Installing the application server

1. On the Type of Installation page, select **Server Installation**.
2. Follow the steps on each page, supplying all required information.

5.4.6 Installing clients

Once the server installation has been done, you can install clients. (No need to do this on the server itself, the client is already installed.)

5.4.6.1 Installing a client

1. On the Type of Installation screen, select **Client Installation**, and click **Next**.
2. On the next screen select **Perform a client installation**, click **Next**.
3. Continue the installation to the end, entering all information required. Note that much of this information was also required when installing the WSS application server.

5.4.7 Installing shortcuts

Once the clients are installed, you need to access them using batch files to pass the correct arguments to each client. Suite Installer can create these for you, and if you wish, create Windows shortcuts, accessible from your Windows **Start** menu.

1. On the Type of Installation screen, select **Client Installation**, and click **Next**.
2. On the next screen select **Add new client shortcut**, and click **Next**, check the information on the next screen and click **Next**.
3. On the Configuration Server screen, enter the machine name of the computer where the Configuration Server is located, and the environment name and number that you are installing clients for.
4. Continue the installation to the end, entering all information required.

Your list of shortcuts will look something like this, depending on what options you installed:

acm

PMM.bat

securitycenter

TRM.bat

TRM_env.bat

TRM_shell.bat

TRM_trusted.bat

websuite

Of course you can edit these names as you wish.

5.4.8 Suite Installer tips

Repeated installation

When you perform an installation, the Suite Installer saves all the parameters that your entered during the process in a file called `config_X` (where X is a number).

This file can be re-used to perform or redo an installation with the following parameter `--config`

```
WSSInstaller.bat --config  C:\install\7.n\SI_Tools\SIWizard\logs\27_07_09_2\config_1
```

Automated installation

If you add the option `--silent`, the installation is performed without the Suite Installer GUI. This can be very useful when the server does not have access to X Server.

```
WSSInstaller.bat --config  C:\install\7.n\SI_Tools\SIWizard\logs\27_07_09_2\config_1
--silent
```

Environment color schemes

You can select one of four color schemes. You choose the initial color scheme and change it later using the Suite Installer. For full details, search for **Environment color schemes** in the *WSS System Admin Guide*.

5.4.9 Other options

Generating a config file when you cannot launch the Suite Installer

If you do not have access to X Server, you cannot launch the Suite Installer GUI. So on a client, you can launch it with the following option:

```
WSSInstaller.bat --debug-simulation --no-verification
```

This generates a `config_X` file from the parameters that you entered. Note that these parameters will not be verified by the Suite Installer.

Now, on the server which has no X Server, launch the command:

```
WSSInstaller.bat --config  <the_config_file_generated_on_client>  --silent
```

Creating and upgrading a database without SDM

If you launch the Suite Installer with the `--no-sdm` option, the Suite Installer and the database scripts assume that you do not want your database to be SDM-enabled.

Warning: This option is not recommended, as all the Wallstreet Suite database must normally be configured with SDM enabled.

5.4.10 Starting the TRM Client (Application Manager)

To start the TRM client, run the command:

```
envs\<env>\bin\trm-<environment>.bat
```

Note: When Application Manager is run for the first time on a client machine, TRM installs some software (Microsoft Visual C++ 2008 Redistributable components: `vcredist_x86.exe` and `vcredist_x64.exe`). This installation requires that the user has administrator privileges, which may be removed after this one-time installation.

Another operation is performed automatically when starting Application Manager: the registration of OLE Control extensions `vcfi32.ocx`, `TeeChart8.ocx`, and `Calendar.ocx`. This is done with `regsvr32` commands in silent mode. If the user does not have administrator privileges, this command does nothing. For details related to this operation see *Can I remove administrator permissions on Windows machines?* on page 176.

5.5 Creating and populating the database

Important: If you asked the Suite Installer to create a database for you, ignore this section.

5.5.1 Introduction

This chapter describes how to create a database (see below), and how to load static data into the database (see *5.7 Loading static data manually* on page 65).

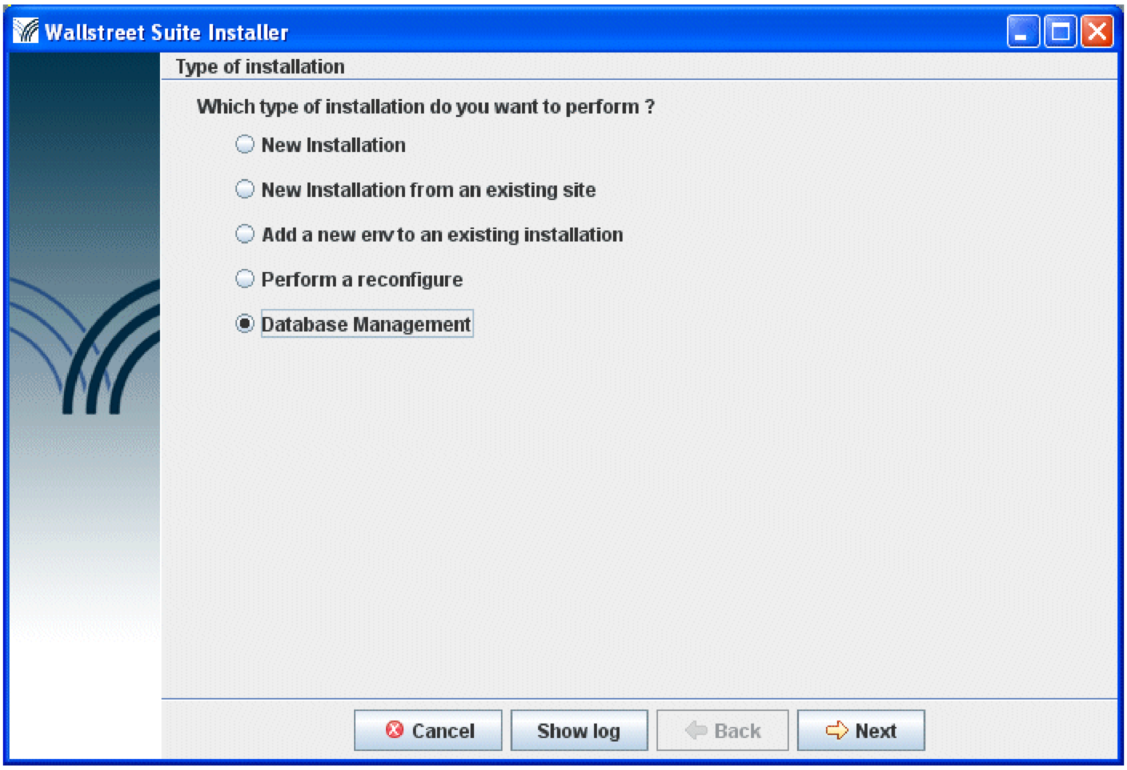
The Suite Installer uses the database properties that are read from your environment to generate script files that it calls to process the database build. These scripts files are generated in the `SIWizard/logs` structure and can be modified or manually launched.

Important: If you want to create and manage an environment without SDM (**not recommended with versions higher than Wallstreet Suite 7.3**), launch the Suite Installer with the option `--no-sdm`.

5.5.2 Performing a build

The Suite Installer uses the database properties that are read from your environment to generate script files that it calls to process the database build. These scripts files are generated in the `SIWizard/logs` structure and can be modified or manually launched.

1. Start the Suite Installer, see *5.3 Expand the Wallstreet Suite package* on page 58, select the **Database Management** option, and click **Next**.



Type of installation

Which type of installation do you want to perform ?

○ New Installation

○ New Installation from an existing site

○ Add a new env to an existing installation

○ Perform a reconfigure

● Database Management

2. On the next screen, select **Database Build**, and click **Next**.
3. Choose and existing environment and click **Next**.
4. Follow the popup help to guide you through the remaining screens.

5.5.3 DBCA

On Oracle, you may need to use DBCA to create and manipulate the database.

You can either:

- Access this with the Suite Installer: **Server installation - Database Management - Expert Mode - Oracle DBCA Build**

Or

- Use the DBCA tool provided with the Oracle installation. The template files to use for database creation are in

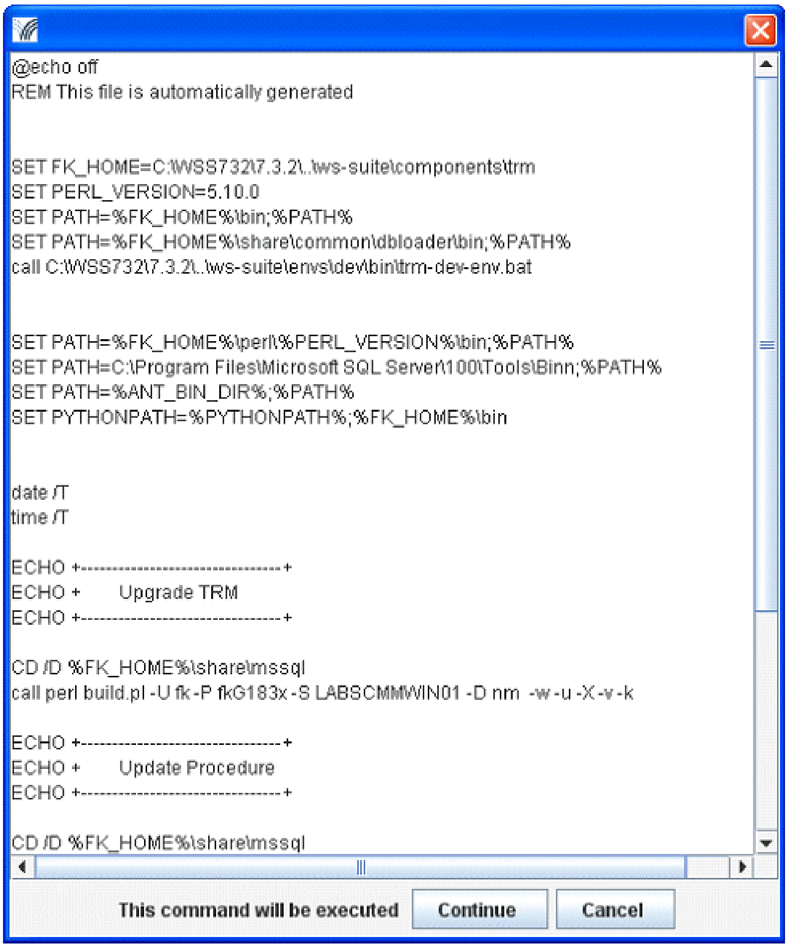
```
SI_Generic\db-build-framework_*.zip\oracle\dbca
```

5.6 Database upgrade using Suite Installer

Database upgrade can also be done using the Suite Installer:

1. Start the Installation wizard, see *5.3 Expand the Wallstreet Suite package* on page 58, and select the **Database Management** option, and click **Next**.
2. On the next screen, select **Database Upgrade**, and click **Next**.

   For each module, the Suite Installer generates an upgrade script and displays it before execution.



```
@echo off
REM This file is automatically generated


SET FK_HOME=C:\WSS732\7.3.2\..\ws-suite\components\trm
SET PERL_VERSION=5.10.0
SET PATH=%FK_HOME%\bin;%PATH%
SET PATH=%FK_HOME%\share\common\dbloader\bin;%PATH%
call C:\WSS732\7.3.2\..\ws-suite\envs\dev\bin\trm-dev-env.bat


SET PATH=%FK_HOME%\perl\%PERL_VERSION%\bin;%PATH%
SET PATH=C:\Program Files\Microsoft SQL Server\100\Tools\Binn;%PATH%
SET PATH=%ANT_BIN_DIR%;%PATH%
SET PYTHONPATH=%PYTHONPATH%;%FK_HOME%\bin


date /T
time /T

ECHO +--------------------------------+
ECHO +       Upgrade TRM
ECHO +--------------------------------+

CD /D %FK_HOME%\share\mssql
call perl build.pl -U fk -P fkG183x -S LABSCMMWIN01 -D nm  -w -u -X -v -k

ECHO +--------------------------------+
ECHO +       Update Procedure
ECHO +--------------------------------+

CD /D %FK_HOME%\share\mssql
```

   Scripts are generated in `SI_Tools/SIWizard/logs/DD_MM_YY_N`.

   For example, on Windows:

   `upgradeACM_1.bat`

   `upgradeTRM_1.bat`

   `upgradeTRM_2.bat`

5.6.1 Upgrade Setup script

After a database upgrade, you should perform a setup upgrade using the Suite Installer.

Important: Before doing such task, you should merge your current setup CSD with the upgrade scripts.

To perform the upgrade, select **Server Installation - Database Management - Setup Upgrade**

(This is the equivalent to `build -U -P -D -v -k -w -E setup/*.sql`)

After running this, a dialog appears where you can verify the script and edit it against the CSD setup.

5.7 Loading static data manually

Static data is loaded automatically by SIWizard. This section describes the manual process for specific needs, and is not part of a normal installation.

5.7.1 Loading default static data

Note: This should be done in a Windows environment.

You will:

1. Unzip the file containing the default static data (in XML format), provided in the Wallstreet Suite distribution.
2. Use the DBLoader tool, also provided in the Wallstreet Suite distribution, to load the XML data into the Wallstreet Suite database.

5.7.1.1 Unzipping the static data file

Navigate to the `..\Factory_Generic\static-data` folder where you downloaded the Wallstreet Suite packages.

Unzip the <factory>.zip file to a suitable temporary location.

5.7.1.2 Using DBLoader

1. Start the shell that corresponds to your environment. For example, if your environment is called `dev`, then start:

   Windows:

   ```
   C:\WSS\ws-suite\envs\dev\bin\trm-dev-shell.ba
   ```

   Unix:

   ```
   /wss/ws-suite/envs/dev/bin/trm-dev-shell.sh
   ```

   This evaluates the environment required to run the DBloader tool.

2. In the same Command Prompt widow, change to the directory where DBLoader is:

   Windows:

   ```
   cd %FK_HOME%\share\common\dbloader\bin
   ```

   Unix:

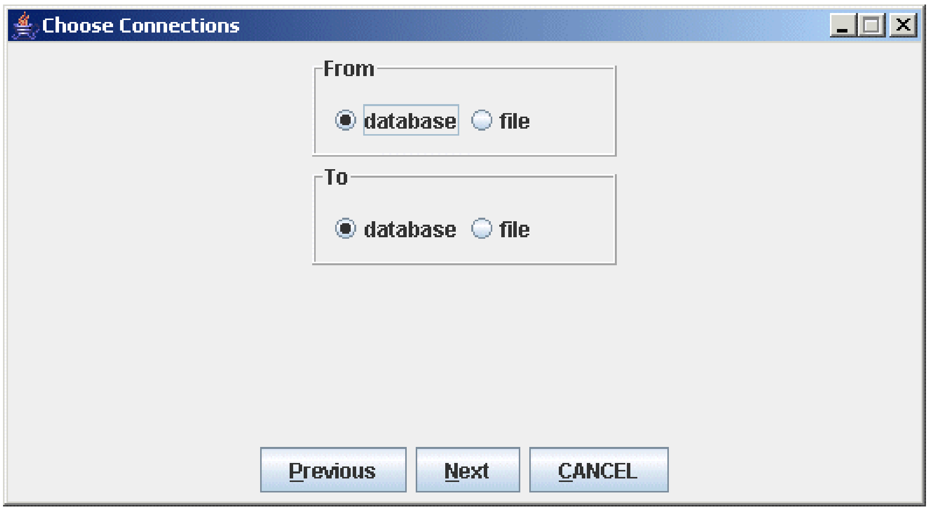
   ```
   cd $FK_HOME/share/common/dbloader/bin
   ```

3. Run this script:

```
Python generic.py
```

Note: To display the following graphical client in a non-Windows environment, you must have a graphical component (X11 on Unix or Linux).

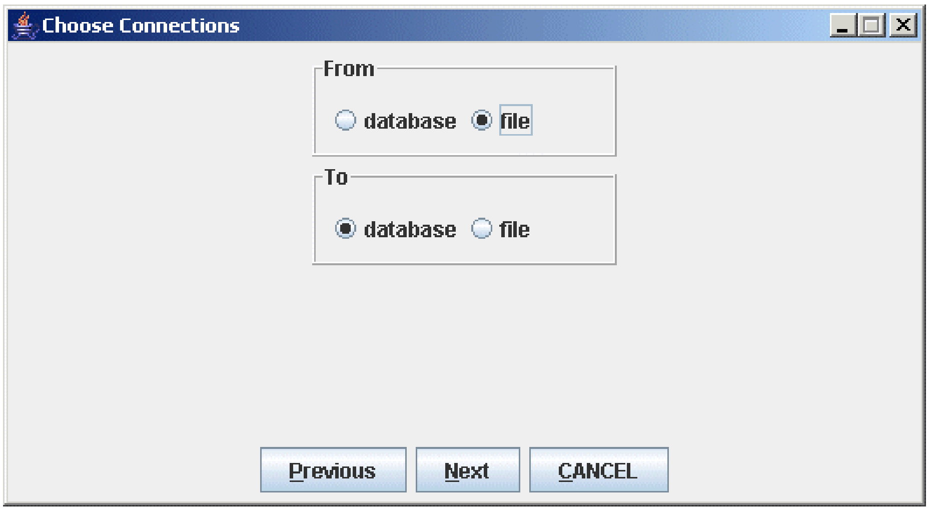
After a few seconds you see this:



From: ● database ○ file

To: ● database ○ file

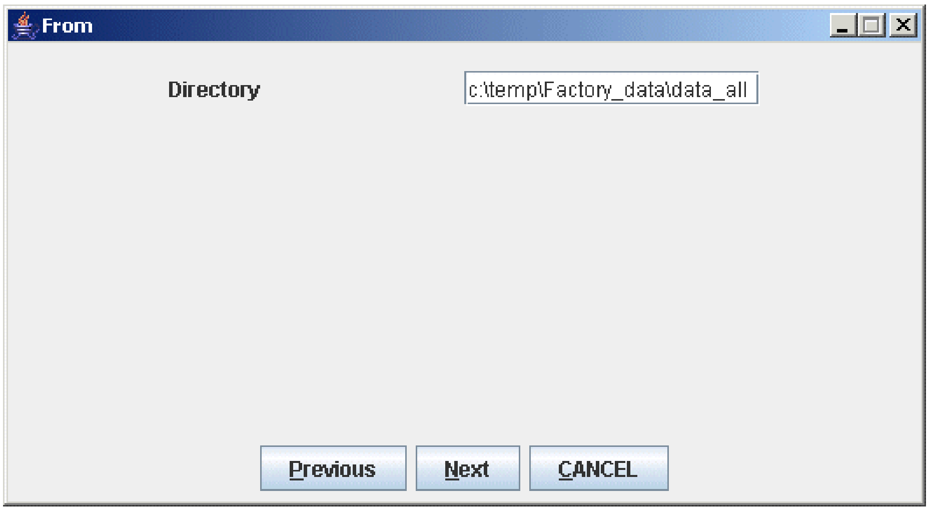
4. Select the **From file** option:



From: ○ database ● file

To: ● database ○ file

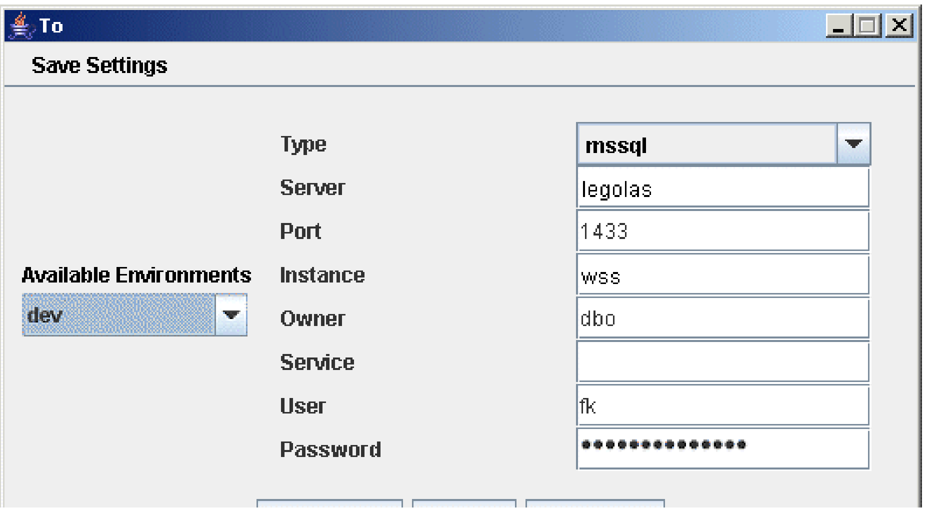
5. Click the **Next** button:d



Directory c:\temp\Factory_data\data_all

In the **Directory** field, enter the path to the XML files you unzipped in *5.7.1.1 Unzipping the static data file* on page 65.

6. Click the **Next** button:



Available Environments dev

Type mssql

Server legolas

Port 1433

Instance wss

Owner dbo

Service

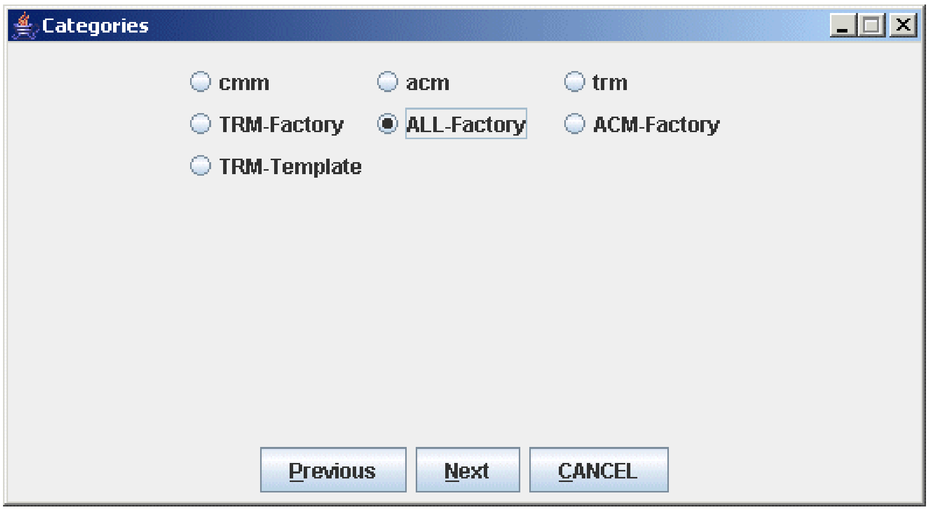
User fk

Password ***************

Enter information about the target database as follows:

Type	Database type: `Oracle` (Oracle), `sqlserver` (MSSQL), `jtds-sybase` (Sybase)
Server	Database server machine name
Port	Database port number
Instance	Oracle: the Oracle System ID (SID) MSSQL and Sybase: the name of the database
Owner	Database owner, usually `DBO` for Oracle, and `dbo` for MSSQL and Sybase
Service	MSSQL only: the MSSQL instance name if more than one database instance exists.
User	User login
Password	User password

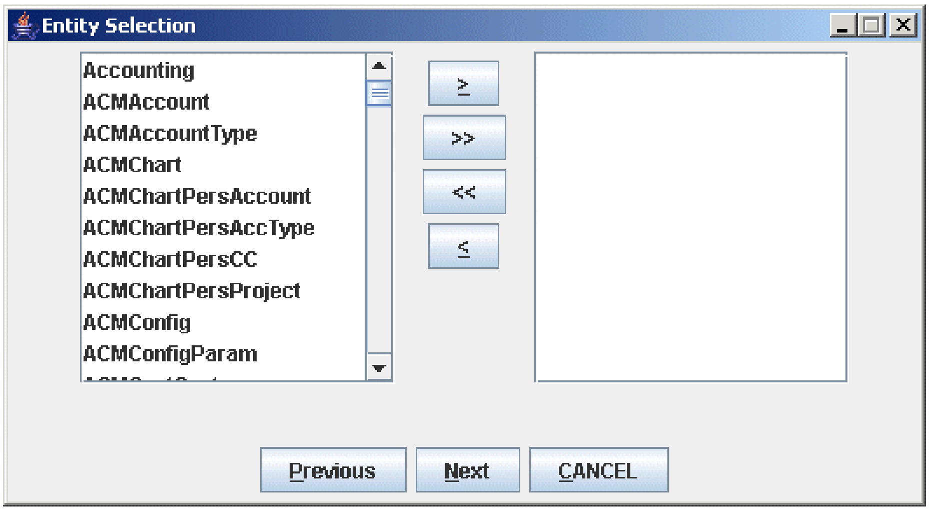
7. Click the **Next** button:



○ cmm ○ acm ○ trm
○ TRM-Factory ● ALL-Factory ○ ACM-Factory
○ TRM-Template

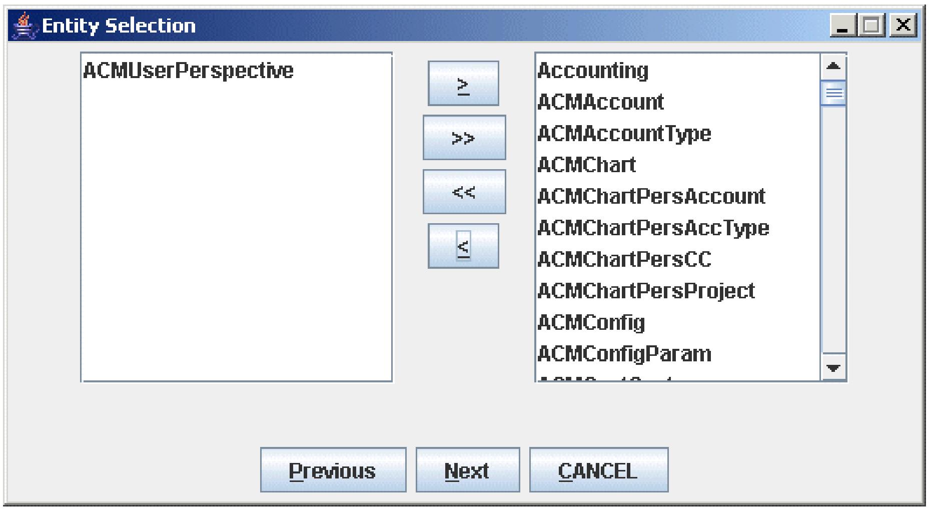
Select the category: for a full set of static data, select **ALL-Factory**. This category contains all objects needed for TRM, ACM, and CMM that have had the data set up by the Factory team.

8. Click the **Next** button:



Accounting
ACMAccount
ACMAccountType
ACMChart
ACMChartPersAccount
ACMChartPersAccType
ACMChartPersCC
ACMChartPersProject
ACMConfig
ACMConfigParam

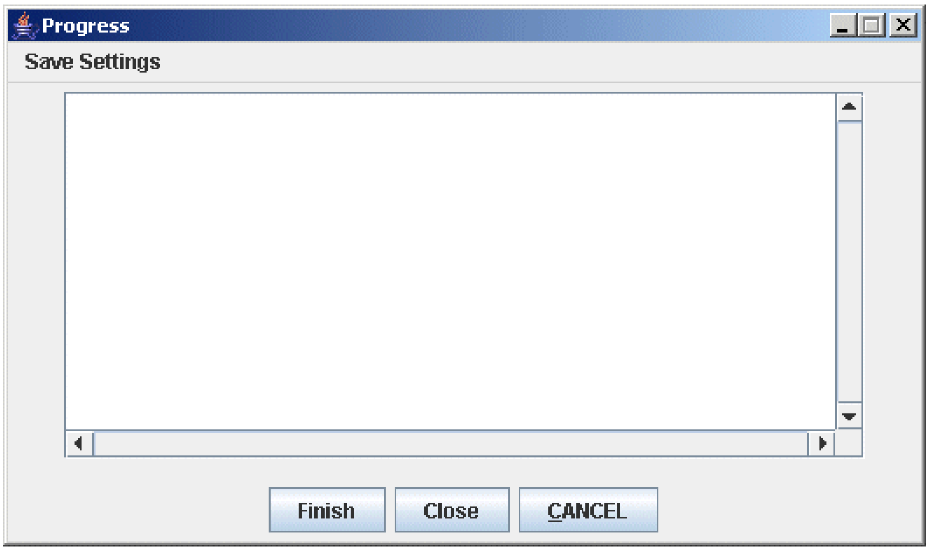
Click the >> button to select all static data entities, then select ACMUSer Perspective and click the < button, see *5.7.1.2.1 Perspective checking* on page 69 before continuing:



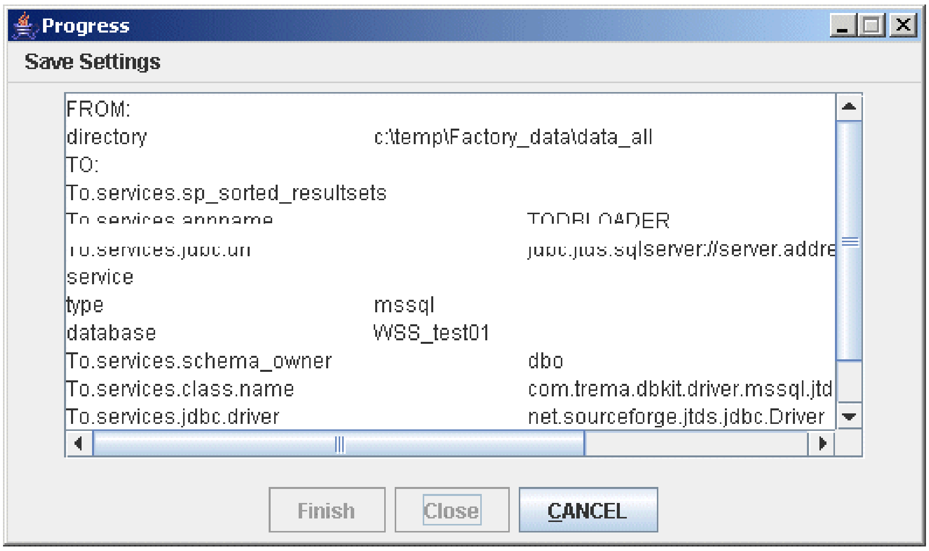
ACMUserPerspective

Accounting
ACMAccount
ACMAccountType
ACMChart
ACMChartPersAccount
ACMChartPersAccType
ACMChartPersCC
ACMChartPersProject
ACMConfig
ACMConfigParam

9. Click the **Next** button:



10. Click the **Finish** button to start the conversion and transfer of XML file data into the Wallstreet Suite database:



```
FROM:
directory                    c:\temp\Factory_data\data_all
TO:
To.services.sp_sorted_resultsets
To.services.appname                         TODBLOADER
To.services.jdbc.url                        jdbc:jtds:sqlserver://server.addre
service
type                         mssql
database                     WSS_test01
To.services.schema_owner                    dbo
To.services.class.name                      com.trema.dbkit.driver.mssql.jtd
To.services.jdbc.driver                     net.sourceforge.jtds.jdbc.Driver
```

Click **Close** to close the DBLoader application.

5.7.1.2.1 Perspective checking

When loading ACM data (see step *8.* on page 68), you must turn off perspective checking. It is a Configuration setup and can be achieved with the following sql:

```
Update Configuration set value='true' where id=10000
```

Alternatively, use the Configuration Table Editor application.

5.7.1.3 The log file

The log file is kept in:

```
components\trm\share\common\dbloader\bin\var\log\dbloader.log
```

5.8 Importing existing static data

This section describes a method that uses Excel workbooks and WSS tools to import existing customer static data into the Wallstreet Suite database.

The file `data-gathering.zip` contains sample Excel worksheets that can currently manage the following static data entities:

- Client
- Client Group
- Domain
- ChartOfAccount and Accounts
- Accounting Period
- Accounting Mapping Rule Info

5.8.1 The process

The process is as follows:

Open the relevant workbook within `data-gathering.zip`, for example `client.xls` for clients, then

1. Verify input data.
2. Verify output data.
3. Save data as CSV files.
4. Convert the CSV files to XML format.
5. Upload the content of the XML files to the database with the DBLoader tool.

5.8.2 Documentation

Each workbook contains an **Instructions** worksheet that describes in detail how use the workbook.

5.8.3 Verifying input data

First verify that all data in the **Input-** worksheets is properly entered, including the following:

- No empty rows in the middle of the input data
- All mandatory fields (column headings in red) have been filled in.

5.8.4 Verifying output data

All workbooks are structured so that **Input-** worksheets are followed by:

- A sheet called **Maps - Lists**
- One or more output sheets.

The worksheet **Maps - Lists** is used to store selection lists used in **Input-** sheets and in some cases to convert user friendly names used in input sheets to database ids (e.g. in Country, to convert "Germany" to "DE"). At the time of data upload nothing needs to be done in this sheet.

All the worksheets following **Maps - Lists** store output data from the workbook. The reason for having separate output sheets is to be able to do some of the conversions discussed above, and to be able to map input data to the corresponding database field thus facilitating an automatic upload.

All output sheets contain formulas that work on **Input-** sheets, and it is very important that these work properly. Make sure never to make any changes in these worksheets (except the ones described below).

If no changes have been made to any of the **Input-** sheets while entering the data, nothing needs to be done in output sheets either. However, if new rows have been added to **Input-** sheets, this may mean that not all input data is included in the output sheets. In order to make sure all data is included, take the following steps:

1. Check if the number of rows in current output range (gray background) is enough to capture all data from corresponding **Input-** sheet. If this is not the case (if you see that the gray area is full) take the following steps:
   a. Go to the bottom of the gray area and add the necessary number of new rows into the gray area (above the current last row)
   b. Go back to the one of the old rows containing the formulas, select that row by clicking the row heading, and copy that row to all rows below it including the last row of the now extended gray area by using the **Paste Special - Formulas** option (this preserves all formatting).
   c. Make sure that you now have all data included in the gray area.
2. Move to next output worksheet and repeat step 1.
3. Repeat the above step for all output worksheets.

5.8.4.1 Saving worksheet data as CSV files

The automatic data upload is based on CSV files saved from the workbook output sheets. For a successful upload, check the following:

- **Every** output worksheet must be saved as a CSV file. For technical reasons, output worksheets with no data in them **must** be saved, or the upload will fail.
- The name of the CSV file must be exactly the same as the name of the worksheet as it appears on the tab (e.g. `ClientAccount.csv` from worksheet **ClientAccount** in the Client data workbook).
- The CSV files must be accessible to the scripts used to convert them to XML files.

To save worksheets as CSV files:

1. Go to the first output worksheet of the workbook. This is the first worksheet after **Maps - Lists**.
2. Select the **File - Save As...** option.
3. In the **Save as type:** dropdown list, select **CSV (Comma delimited)**.
4. In the **Save in:** dropdown list, navigate to the folder where you keep CSV files.
5. Verify that the *name*.csv in the **File name:**" field is the name of the worksheet you are saving.
6. Click the **Save** button.

   Excel displays a dialogue asking if you only want to save the active sheet. Click the **OK** button.

   Excel displays another dialogue warning about features in the file not supported by the CSV format, and asks if you want to keep the format of the workbook. Click the **Yes** button.
7. Move to the next output worksheet, and repeat steps 2 to 8.
8. Repeat the above steps until all output worksheets have been saved as CSV files (including ones with no data).

5.8.4.2 Converting the CSV files to XML

To upload to the database, the CSV files must be converted to an XML format that is suitable for the use with the DBLoader tool. The file `csv-loader.zip` contains the perl scripts and shell scripts to do the conversions. You can find this file in the `..\ALL_Modules_Factory\tools` folder.

This zip file should be unpacked in a suitable location where a perl language interpreter is available (for example the machine where Wallstreet Suite is installed). Follow these steps:

1. Ensure that all CSV files are available in one directory (for example `c:\tmp\csvdirectory`).
2. Create a directory for storing the XML files (for example `c:\tmp\xmldirectory`).
3. Start a Windows Command Prompt session, or a bash shell on Unix.
4. Change directory (cd) to the `bin` directory.

5. Run this command:

   Windows: `run.bat c:\tmp\csvdirectory c:\tmp\xmldirectory`

   Unix: `sh run.sh /tmp/csvdirectory /tmp/xmldirectory`

Note: The Client Editor in TRM allows a client ID with non-alphanumeric characters, for example "P & N". This is not acceptable to CMM.

The conversion performs checks on the client ID, and if there are any unacceptable characters used, then a message is displayed, and the conversion process fails. If you do not use CMM, and you wish to override this, add a third parameter - 1 - to the relevant command above. For example:

```
sh run.sh /tmp/csvdirectory /tmp/xmldirectory 1
```

5.8.4.3 Uploading the XML files to the database using DBLoader

To upload the XML files, you use the DBLoader tool.

5.8.4.3.1 DBLoader under Windows

Follow the steps starting at *5.7.1.2 Using DBLoader* on page 65, using the path where your XML files are stored.

5.8.4.3.2 DBLoader under Unix

Set up properties files for both the database connection and the directory information and then use the program `file-to-db-nowin.sh`, using the options `file.properties db.properties All-Factory.category`.

Properties files: contents

File name	Properties with example values
file.properties	`Directory=/tmp/xmldirectory`

db.properties	`Server=LABS-PC265.corp.trema.com`
	`Port=1433`
	`Owner=DBO`
	`Instance=DBName`
	`User=fk`
	`Password=xxxxxx`

Chapter 6 **Managing Wallstreet Suite processes**

Now that the installation is successfully completed, this chapter describes how you work with your new installation's real time processes.

6.1 Different methods of managing Wallstreet Suite processes

You can choose among these alternatives:

1. Use the Windows-only application pm_admin.exe ("Process Monitor") supplied in the Wallstreet Suite package. This provides a graphical, flexible, and powerful way of monitoring and controlling Wallstreet Suite processes, both Windows and Unix versions. This is the method described in this chapter, and is the one recommended by Wallstreet Systems.
2. Use the Windows or Unix application pm_cmd (.exe/.sh) which provides a command line version of Process Monitor.
3. Unix only: use the legacy rc scripts (modified to work with the environment used by Suite Installer), which you can then incorporate into your own third-party or bespoke process management system. These are in the directory `<installation>/envs/<environment>/etc`:

    rc.comkit

    rc.limitd

    rc.mdsd

    rc.names

    rc.sdm

    rc.encrypt

    rc.mirror

    rc.onyx

    rc.trmswift

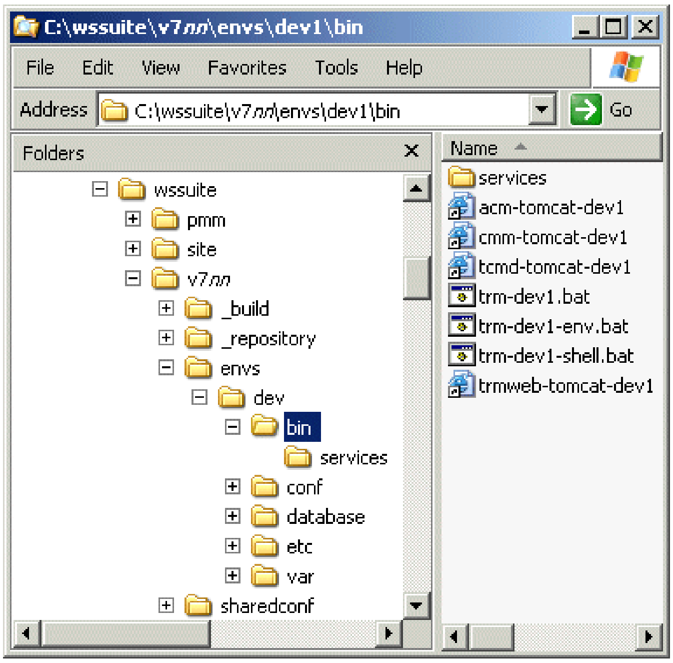
6.2 Quick tour of the installation

Your new installation on the Application Server should look similar to the one shown here. If you used the command-line installer, find the installation in the path you gave the `-i` option of the `installpackages` script. If you used the wizard-style installer, you can find it in the path you gave as the **Target directory**.

For each environment there is a `bin` sub folder that contains three important items:

1. Shortcut URLs to connect to each of the J2EE application servers.
2. Environment files used to launch TRM, a TRM Shell, or evaluate a TRM environment.
3. A `services` folder that contains all the command lines used by Process Monitor to start part of the Wallstreet Suite real-time processes.

Example:



6.3 Using Process Monitor

Note: For a full description of Process Monitor, see the *Wallstreet Suite System Administration Guide*.

Wallstreet Suite requires real-time processes to be managed on one or more Wallstreet Suite Application Servers. The application that does this is called Process Monitor. Process Monitor is available either as a command-line interface (available for Windows and Unix) or a graphical interface (Windows only).

Note: You can use either version of Process Monitor to manage Application Servers running Windows or Unix real-time processes.

This chapter describes the Windows version of Process Monitor only.

6.3.1 Launching Process Monitor

Process Monitor consists of three main components:

1. A Process Monitor Agent Daemon (**pmad**). This daemon is used to start and stop the real-time services.
2. A Process Monitor Service Daemon (**pmsd**). This daemon reads the configuration files and sends the start and stop commands to one or more **pmad**s.
3. The Process Monitor application. This connects to a running **pmsd**, and is used to start and stop the Wallstreet Suite services.

Note: **pmsd** and **pmad** can be configured to run as processes or Windows services. By default, the Suite Installer configures them as Windows services when running under Microsoft Windows.

6.3.1.1 Automatically starting up processes

If you want all Wallstreet Suite processes that are flagged as "auto-start" to be started when **pmsd** is launched (either as a process or a Windows Service), do the following:

1. Open the file `%INSTALLATION_HOME%\sharedconf\components\pmm\config.properties`, and ensure that the following line appears:
   - If **pmsd** is to be run as a process: `pmm.process.auto_start=true`
   - If **pmsd** is to be run as a service: `pmm.pmsd.service.process.auto_start=true`
2. Run the Suite Installer `reconfigure` command. Here is an example:

   ```
   %si% reconfigure -i installation_directory -s site_directory -e env_name -n
   env_number
   ```

6.3.1.2 Customizing real-time services from the site directory

The Process Monitor configuration file that describes the available services is in the directory called `%INSTALLATION_HOME%\..\pmm\etc\conf`. The file itself is called `env_`*`env_name_and_number`*`.xml`.

This file is the concatenation of individual template files available under <*installation directory*>`\..\v72\envs\`*`dev`*`\etc\pmm\_modules`. Each of these files describes a particular process.

Each template can be customized (or duplicated to add a new server) after being integrated in the `site` directory. To create a PMM customized template under the `site` directory, do the following:

1. In the `site` directory, create the directory for PMM customized templates `etc\csd\pmm\pmm_templates\conf`
2. Copy the reference templates to customize from `%SI_HOME%\..\v72\envs\`*`dev`*`\etc\pmm\_modules` to <*site directory*>`\etc\csd\pmm\pmm_templates\conf`
3. Customize the templates by editing them. When they are combined into the single `env_`*`env_name_and_number`*`.xml` file, the order in which they appear is determined by the original file's prefix number: you can change the file prefix number to move the file's data up or down the final `env_`*`env_name_and_number`*`.xml` file.
4. Run the `site` command with the `-w` option to trigger a reconfigure:
   - **Windows:**
     ```
     %si% site -s <site directory> -w -i <installation directory> -e env -n 1
     ```
   - **Unix:**
     ```
     $si site -s <site directory> -w -i <installation directory> -e env -n 1
     ```

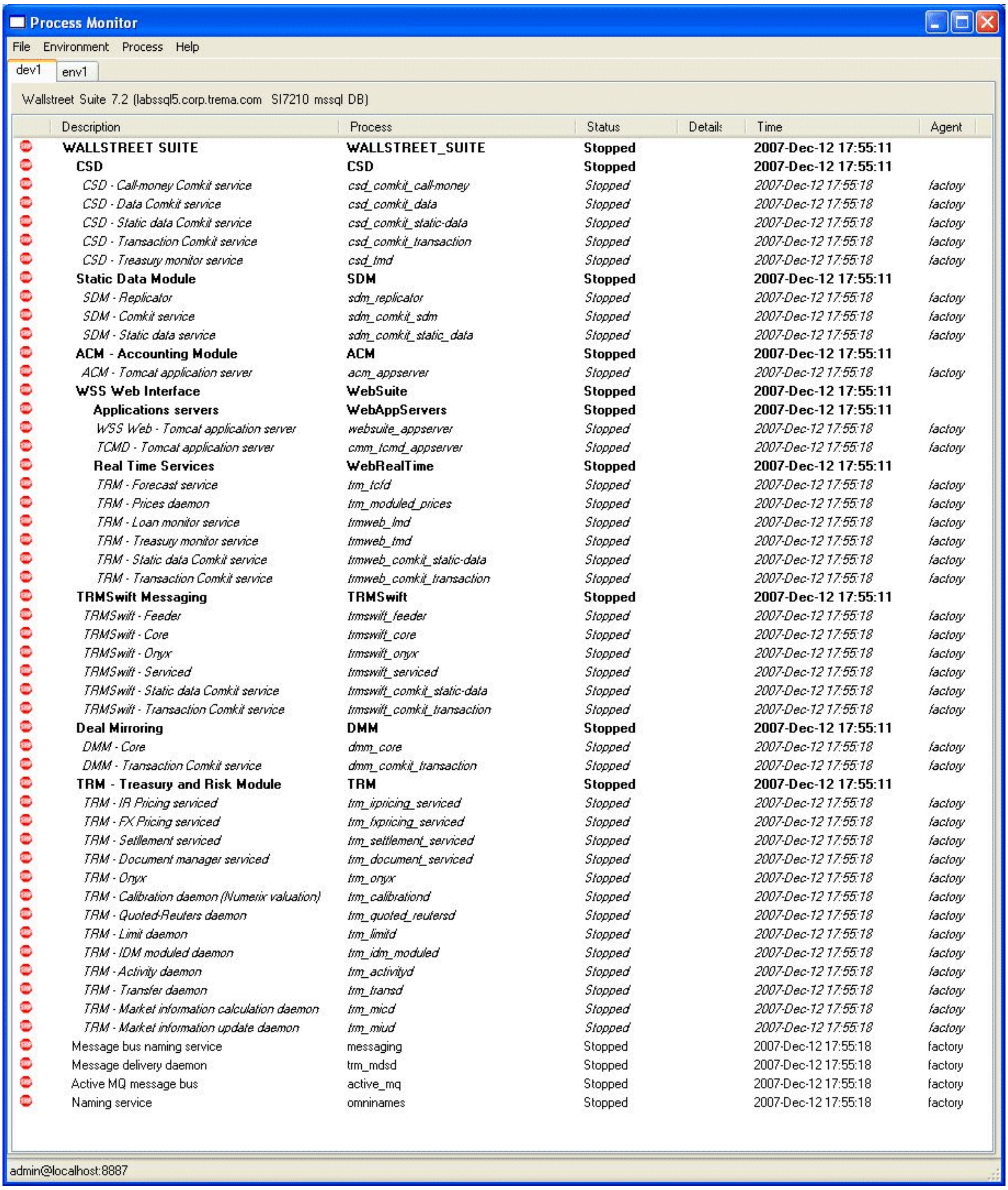
6.3.1.3 Launching pmsd and pmad

By default, **pmsd** and **pmad** are set to run in debug mode. This is to make it easier to troubleshoot installation issues. (The debug mode can be turned off.)

1. Navigate to the `pmm` folder in your installation.
2. Double click `pmad_debug.bat`.
3. Double click `pmsd_debug.bat`.
4. Check the screens to see if any error messages are logged

6.3.2 Launching the Process Monitor application

Navigate to the `pmm` folder and double click the file <hostname>`_admin.bat`. The default user and password for the Process Monitor application are: `admin` and `password`



Wallstreet Suite 7.2 (labssql5.corp.trema.com SI7210 mssql DB)

Description	Process	Status	Details	Time	Agent
WALLSTREET SUITE	**WALLSTREET_SUITE**	**Stopped**		**2007-Dec-12 17:55:11**	
CSD	**CSD**	**Stopped**		**2007-Dec-12 17:55:11**	
CSD - Call-money Comkit service	csd_comkit_call-money	Stopped		2007-Dec-12 17:55:18	factory
CSD - Data Comkit service	csd_comkit_data	Stopped		2007-Dec-12 17:55:18	factory
CSD - Static data Comkit service	csd_comkit_static-data	Stopped		2007-Dec-12 17:55:18	factory
CSD - Transaction Comkit service	csd_comkit_transaction	Stopped		2007-Dec-12 17:55:18	factory
CSD - Treasury monitor service	csd_tmd	Stopped		2007-Dec-12 17:55:18	factory
Static Data Module	**SDM**	**Stopped**		**2007-Dec-12 17:55:11**	
SDM - Replicator	sdm_replicator	Stopped		2007-Dec-12 17:55:18	factory
SDM - Comkit service	sdm_comkit_sdm	Stopped		2007-Dec-12 17:55:18	factory
SDM - Static data service	sdm_comkit_static_data	Stopped		2007-Dec-12 17:55:18	factory
ACM - Accounting Module	**ACM**	**Stopped**		**2007-Dec-12 17:55:11**	
ACM - Tomcat application server	acm_appserver	Stopped		2007-Dec-12 17:55:18	factory
WSS Web Interface	**WebSuite**	**Stopped**		**2007-Dec-12 17:55:11**	
Applications servers	**WebAppServers**	**Stopped**		**2007-Dec-12 17:55:11**	
WSS Web - Tomcat application server	websuite_appserver	Stopped		2007-Dec-12 17:55:18	factory
TCMD - Tomcat application server	cmm_tcmd_appserver	Stopped		2007-Dec-12 17:55:18	factory
Real Time Services	**WebRealTime**	**Stopped**		**2007-Dec-12 17:55:11**	
TRM - Forecast service	trm_tcfd	Stopped		2007-Dec-12 17:55:18	factory
TRM - Prices daemon	trm_moduled_prices	Stopped		2007-Dec-12 17:55:18	factory
TRM - Loan monitor service	trmweb_lmd	Stopped		2007-Dec-12 17:55:18	factory
TRM - Treasury monitor service	trmweb_tmd	Stopped		2007-Dec-12 17:55:18	factory
TRM - Static data Comkit service	trmweb_comkit_static-data	Stopped		2007-Dec-12 17:55:18	factory
TRM - Transaction Comkit service	trmweb_comkit_transaction	Stopped		2007-Dec-12 17:55:18	factory
TRMSwift Messaging	**TRMSwift**	**Stopped**		**2007-Dec-12 17:55:11**	
TRMSwift - Feeder	trmswift_feeder	Stopped		2007-Dec-12 17:55:18	factory
TRMSwift - Core	trmswift_core	Stopped		2007-Dec-12 17:55:18	factory
TRMSwift - Onyx	trmswift_onyx	Stopped		2007-Dec-12 17:55:18	factory
TRMSwift - Serviced	trmswift_serviced	Stopped		2007-Dec-12 17:55:18	factory
TRMSwift - Static data Comkit service	trmswift_comkit_static-data	Stopped		2007-Dec-12 17:55:18	factory
TRMSwift - Transaction Comkit service	trmswift_comkit_transaction	Stopped		2007-Dec-12 17:55:18	factory
Deal Mirroring	**DMM**	**Stopped**		**2007-Dec-12 17:55:11**	
DMM - Core	dmm_core	Stopped		2007-Dec-12 17:55:18	factory
DMM - Transaction Comkit service	dmm_comkit_transaction	Stopped		2007-Dec-12 17:55:18	factory
TRM - Treasury and Risk Module	**TRM**	**Stopped**		**2007-Dec-12 17:55:11**	
TRM - IR Pricing serviced	trm_irpricing_serviced	Stopped		2007-Dec-12 17:55:18	factory
TRM - FX Pricing serviced	trm_fxpricing_serviced	Stopped		2007-Dec-12 17:55:18	factory
TRM - Settlement serviced	trm_settlement_serviced	Stopped		2007-Dec-12 17:55:18	factory
TRM - Document manager serviced	trm_document_serviced	Stopped		2007-Dec-12 17:55:18	factory
TRM - Onyx	trm_onyx	Stopped		2007-Dec-12 17:55:18	factory
TRM - Calibration daemon (Numerix valuation)	trm_calibrationd	Stopped		2007-Dec-12 17:55:18	factory
TRM - Quoted-Reuters daemon	trm_quoted_reutersd	Stopped		2007-Dec-12 17:55:18	factory
TRM - Limit daemon	trm_limitd	Stopped		2007-Dec-12 17:55:18	factory
TRM - IDM moduled daemon	trm_idm_moduled	Stopped		2007-Dec-12 17:55:18	factory
TRM - Activity daemon	trm_activityd	Stopped		2007-Dec-12 17:55:18	factory
TRM - Transfer daemon	trm_transd	Stopped		2007-Dec-12 17:55:18	factory
TRM - Market information calculation daemon	trm_micd	Stopped		2007-Dec-12 17:55:18	factory
TRM - Market information update daemon	trm_miud	Stopped		2007-Dec-12 17:55:18	factory
Message bus naming service	messaging	Stopped		2007-Dec-12 17:55:18	factory
Message delivery daemon	trm_mdsd	Stopped		2007-Dec-12 17:55:18	factory
Active MQ message bus	active_mq	Stopped		2007-Dec-12 17:55:18	factory
Naming service	omninames	Stopped		2007-Dec-12 17:55:18	factory

admin@localhost:8887

6.3.3 Managing the Services

Process Monitor can be used to start and stop Wallstreet Suite services.

The **Description** column lists services and service groups. Service groups are shown in **bold**, and underneath them (not in bold) are the members of the group.

If you right-click a group and select the **Dependencies** option, you see a list of dependent services which include the members of the group and possibly others too.

If you right-click a group and select **Start**, you start all dependent services. For example, to start all Suite processes, select **WALLSTREET SUITE** and click **Start**.

Services displayed in italics can only be started from the service group. However they can be killed.

Note: The base base services are **Naming service**, **Message bus naming service**, **Active MQ** and **Message delivery daemon**. These must be stopped manually.

6.3.3.1 Debugging

6.3.3.1.1 Process Monitor

Process Monitor has a debug mode, set in `\sharedconf\components\pmm\config.properties` (shown below turned off):

```
# Startup debugging
#pmm.debug.param=--debug
pmm.debug.param=--nodebug
```

When turned on, stopping or killing a process leaves the process window open.

6.3.3.1.2 pmsd and pmad

These two daemons have a `--debug` option for running in debug mode. If you run them without this option, they run silently.

6.4 Additional Process Monitor information

For additional information about Process Monitor, see the *Wallstreet Suite System Administration Guide*.

6.5 Application Server checks

Verify that WebSuite is running by opening a Web browser and navigating to `http(s)://${appserver.hostname}:${ts.appserver.t1.http.port}/websuite`

On Weblogic, navigate to `http://${hostname}:${portnumber}/websuite/start`

Verify that TCMD is running by opening a Web browser and navigating to `http(s)://${appserver.hostname}:${ts.appserver.t2.http.port}/tcmd`

Verify that ACM is running by opening a Web browser and navigating to: `http(s)://${appserver.hostname}:${ts.appserver.t3.http.port}/acm`

6.6 Logging

6.6.1 Log files

The main place to look for log files is in the directory `envs/<environment>/var/logs`

6.6.1.1 Exceptions

- CMM log directories:

  `envs/<environment>/etc/wss-web/cmm/Runtime/var`

  `envs/<environment>/etc/wss-web/cmm/VirtualDirectory/logs`

- TCMD log directories:

  `envs/<environment>/etc/tcmd-weblogic/InstallationData/installation/logging`

  or

  `envs/<environment>/etc/tcmd-tomcat/InstallationData/installation/logging`

- Tomcat log directories:

  `envs/<environment>/var/appservers/websuite-tomcat/logs`

  `envs/<environment>/var/appservers/tcmd-tomcat/logs`

  `envs/<environment>/var/appservers/acm-tomcat/logs`

- Weblogic admin log file:

  ```
  envs/<environment>/var/appservers/weblogic/<domain
  name>-domain/servers/AdminServer/logs/AdminServer.log
  ```

6.6.2 Log configuration

6.6.2.1 TRM realtime processes

The environment variables `DATABASE_DEBUG` and `FK_TRACE_LEVEL` manage the log level of TRM realtime processes. For troubleshooting, it is recommended that you use these values:

Windows:

```
> set DATABASE_DEBUG=1

> set FK_TRACE_LEVEL=30
```

Unix:

```
> DATABASE_DEBUG=1

> export DATABASE_DEBUG

> FK_TRACE_LEVEL=30

> export FK_TRACE_LEVEL
```

Allowable values are: `DATABASE_DEBUG` from `1` to `3`, and `FK_TRACE_LEVEL` from `1` to `99`.

6.6.2.2 Websuite

TRMWeb log level can be controlled through the following `log4j` configuration file:
`envs\<environment>\etc\wss-web\trmweb\conf\log4j\log4j-application-server.xml`

6.6.2.3 SDM

The SDM log level can be controlled through the following log4j configuration file:
`envs\<environment>\etc\sdm\log4j.properties`

6.6.2.4 ACM

The ACM log level can be controlled through the following log4j configuration file: `envs\<environment>\etc\acm\acm-log4j-<environment>.properties`

6.6.2.5 TRMSwift

the TRMSwift log level can be controlled through a set of log4j XML configuration files under %`FK_HOME%\etc\trmswift`:

- `log4j.archive.mmo.xml`
- `log4j.archive.xml`
- `log4j.core.xml`
- `log4j.debug.xml`
- `log4j.feeder.xml`
  `log4j.migrate.xml`
- `log4j.mmo.xml`
- `log4j.smo.xml`
- `log4j.sqlfiles.xml`
- `log4j.update.xml`
- `log4j.stop.xml.`

6.6.2.6 ActiveMQ

The ActiveMQ log level can be controlled through the following log4j configuration file: `envs\<environment>\var\active-mq\conf\log4j.properties`

Chapter 7 Managing WSS modules

7.1 Installing the SDM Synchronizer

The SDM Synchronizer is required when interfacing CMM to TRM. The Synchronizer ensures that static data held in CMM is synchronized with any changes and additions made to static data held in TRM.

The Synchronizer installation package is delivered with the TRM package.

7.1.1 Setting up prerequisites

Before you can use the Synchronizer, a valid TRM client distribution must be installed and an SDM-enabled TRM database must be available at runtime.

In the TRM static data editors, there are two tabs which contain static data that are required for CMM but not for other modules:

- Additional Client Information (in the Client Editor)
- Additional Country Information (in the Country Editor).

In a standalone TRM installation, these two tabs are not visible. Therefore, if your organization has migrated from a standalone TRM installation to one that has SDM enabled and that is integrated with CMM, these two tabs will be blank and you will need to manually enter static data in them before transferring client and country entities to CMM through the Synchronizer.

The Synchronizer configuration files are in `$FK_HOME/etc/sdm/`. In a typical setup, you do not need to change these files as they depend on TRM environment variables such as `FK_IDENT` and `CMM_HTTP_ADDR`.

7.1.2 Launching the Synchronizer

1. Before launching the Synchronizer, evaluate the current TRM environment:
   - The `CMM_HTTP_ADDR` environment variable must be set to the URL of the CMM server.
   - The `FK_MODULES` environment variable must be set to `CMM SDM` for each client to allow the client to access SDM editors.
2. The following processes must be started from the TRM environment:
   - Corba naming service
   - Message Delivery Service (mdsd)
   - comKIT SDM static data service

     In Solaris, the command line to start up this service is:

     ```
     $FK_HOME/etc/rc.comkitd sdm
     ```

Finally, the CMM server must be up and running to receive updates from the Synchronizer.

To launch the Synchronizer, launch `rc.sdm` from `$FK_HOME/etc` (in Solaris) or `sdm.bat` from `%FK_HOME%\bin` (in Windows).

7.1.3 Batch synchronizing after installing or upgrading CMM

This procedure is useful when you need to synchronize certain static data entities to CMM (in case data in TRM and CMM is inconsistent for some reason). The following example shows how to synchronize all clients (parties) to CMM in a Unix environment:

1. Bring all entities to the FINAL state.

   In batch mode, the Synchronizer transfers entities in FINAL state and it does not change the state of these entities. In normal mode, by comparison, the Synchronizer transfers entities in SEND state and changes those entities states to either FINAL or FAILED.

2. Ensure SDM ComKIT is running.

3. From the `$FK_HOME/etc` directory, type:

   ```
   rc.sdm -batch -entity_type Client -main_entity_only
   ```

4. When the command has finished without errors, type:

   ```
   rc.sdm -batch -entity_type Client
   ```

Note: In Windows environments, use `%FK_HOME%\bin\sdm.bat` instead of `rc.sdm`. The command results are displayed in the Command Prompt window.

When you do not use the command line argument `-main_entity_only`, the batch process handles dependent entities in addition to main entities. However, as shown in steps 3 and 4 of the example above, you can only do this after all main entities have been sent to CMM. Therefore, you must run the `rc.sdm` (or `sdm.bat`) file twice:

- The first time for main entities only
- The second time for both main and dependent entities.

In addition to the three command line options used in the above example (`-batch`, `-entity_type`, and `-main_entity_only`), there is a fourth command line argument:
`-sdm_id <SDM_ID_OF_ENTITY>` which allows you to send to CMM only one entity with a given ID.

7.1.3.1 Getting help on the sdm command

Type:

```
rc.sdm -help
```

To see command line options help and examples.

7.1.4 Installing SDM as a Windows service

After installing the Synchronizer and checking that it is running properly in a command shell window, you can register it as a Windows service.

Open a command prompt and evaluate the TRM environment. Go to `%FK_HOME%\etc\sdm\windows_service` and launch `install.bat`. **SDM Synchronizer** should then display in the **Services** window (available in the **Administrative Tools** control panel item in most Windows environments).

Use `remove.bat` to unregister the service. You must reinstall the service each time you upgrade or change the service configuration.

Service parameters are in the `wrapper.conf` file. Uncommenting the `wrapper.syslog.loglevel=DEBUG` sends diagnostics to the Windows Event Viewer that are useful for debugging service startup problems.

Note: The `LOCAL_SERVICE` user account must have write permissions on the `%FK_HOME%\var\log\sdm` directory or the synchronizer cannot log errors.

7.2 CMM: configuring the Web server

How you configure the Web server to work with CMM depends on several factors, including the CMM application server, whether the application server and Web server reside on the same server or different servers, and whether the Web server is inside or outside of your organization’s firewall.

The remainder of this chapter presents an example configuration in which:

- The application server is BEA WebLogic
- The Web server is Microsoft Internet Information Services (IIS)
- The application server and Web server reside on the same server.
- The Web server is inside your organization’s firewall.

Note: Wall Street Systems recommends you refer to your application server vendor’s Web site for detailed instructions on configuring the Web server.

7.2.1 Configuring IIS for BEA WebLogic

To configure IIS for BEA WebLogic:

1. In `${iis_home}`, create a new folder called `scripts`.
2. Copy the `iisproxy.dll` and `iisforward.dll` files from `${bea_home}\wlserver_10.0\server\bin` to `${iis_home}\scripts`.
3. In `${iis_home}\scripts`, create a file called `iisproxy.ini`.
4. Open the file in a text editor.
5. Enter the following text:

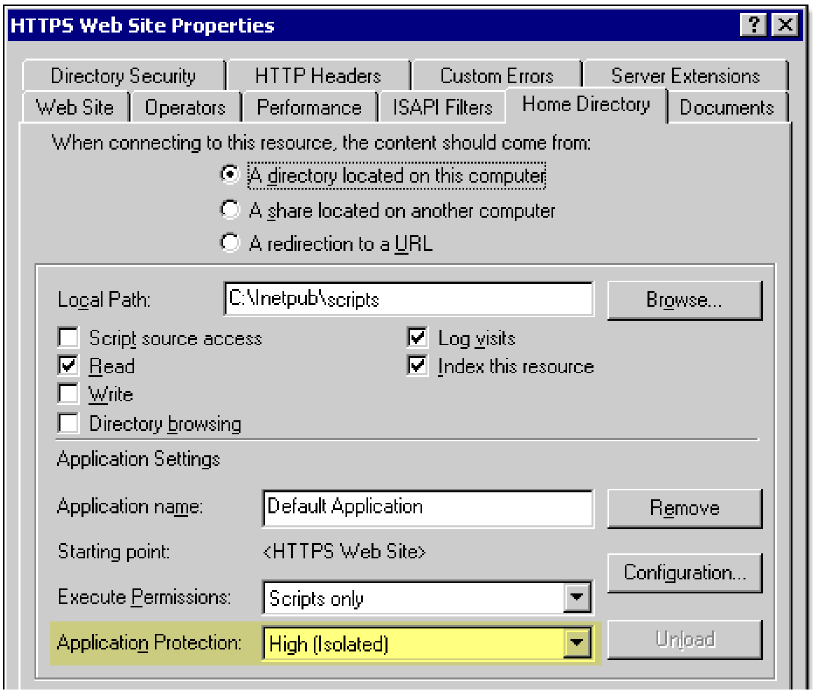
   ```
   WebLogicHost=servername
   WebLogicPort=4001
   WlForwardPath=/websuite
   HungServerRecoverSecs=86400
   Idempotent=OFF
   Debug=ALL
   ```

   Replace `servername` with the actual name of the server.

   The `Debug` flag is for troubleshooting purposes and is usually omitted in production environments.
6. Open IIS Manager.
7. Select and right-click **Default Web Site**.
8. In the resulting menu, select **Properties**.

   The examples in the remainder of this section use a default Web site called "HTTPS Web Site".
9. In the **HTTPS Web Site Properties** dialog, open the **Home Directory** tab.
10. Do one of the following:

– In Windows 2000, ensure the **Application Protection** list is set to `High (Isolated)`:



When connecting to this resource, the content should come from:

● A directory located on this computer
○ A share located on another computer
○ A redirection to a URL

Local Path: C:\Inetpub\scripts

□ Script source access
☑ Read
□ Write
□ Directory browsing
☑ Log visits
☑ Index this resource

Application Settings

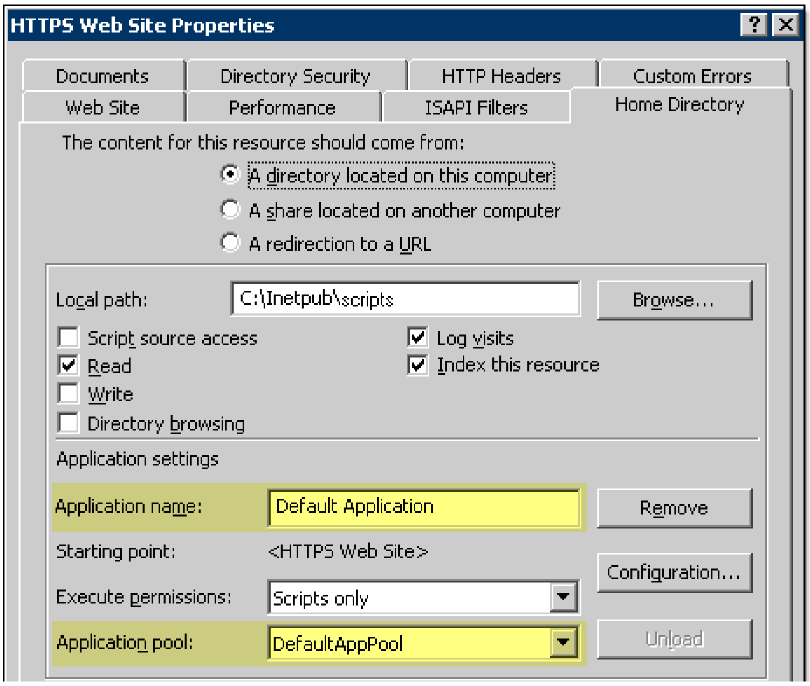
Application name: Default Application

Starting point: <HTTPS Web Site>

Execute Permissions: Scripts only

Application Protection: High (Isolated)

– In Windows Server 2003, ensure the **Application name** field and **Application pool** list contain correct values:Z



The content for this resource should come from:

● A directory located on this computer
○ A share located on another computer
○ A redirection to a URL

Local path: C:\Inetpub\scripts

□ Script source access
☑ Read
□ Write
□ Directory browsing
☑ Log visits
☑ Index this resource

Application settings

Application name: Default Application

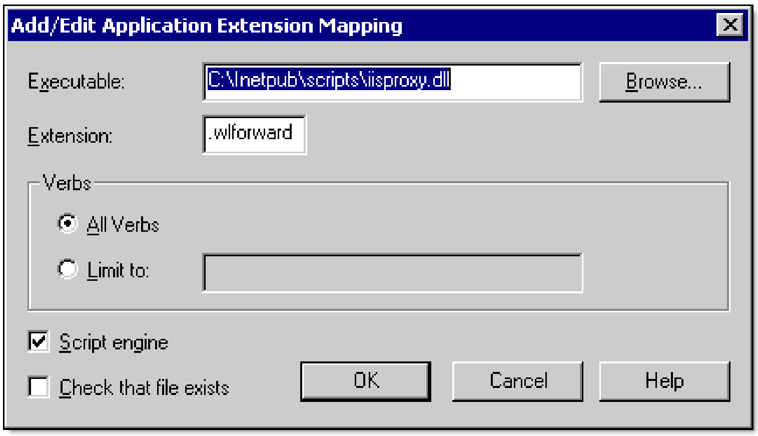
Starting point: <HTTPS Web Site>

Execute permissions: Scripts only

Application pool: DefaultAppPool

11. Click **Configuration...**.

12. In the **Add/Edit Application Extension Mapping** dialog, add the following application extension mapping:



Executable: C:\Inetpub\scripts\iisproxy.dll

Extension: .wlforward

● All Verbs

○ Limit to:

☑ Script engine

□ Check that file exists

Ensure the **Check that file exists** checkbox is cleared.

13. Click **OK**.

14. In the **HTTPS Web Site Properties** dialog, open the **ISAPI Filters** tab.

15. Click **Add...**.

16. In the **Filter Properties** dialog, click **OK**.

17. In the **HTTPS Web Site Properties** dialog, click **Apply**.

18. Click **OK**.

19. :In Windows 2000, restart IIS; In Windows Server 2003:

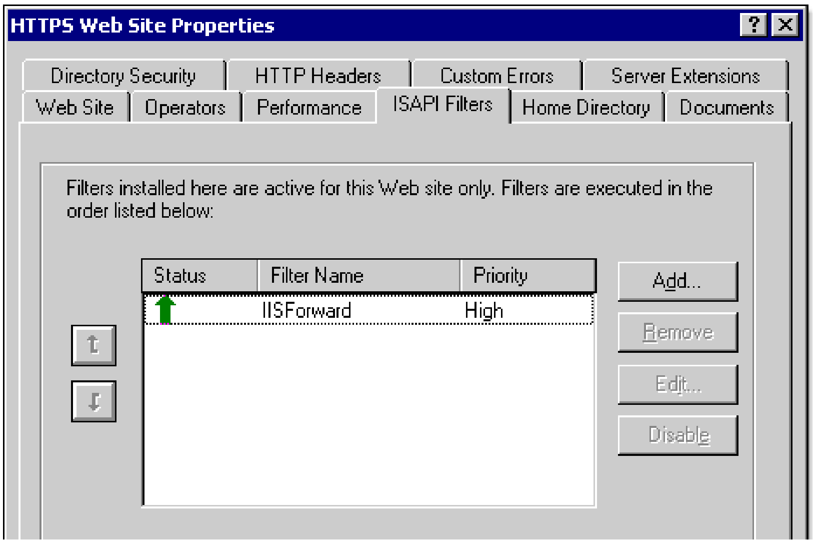
- **a.** Select **Web Service Extensions**.
- **b.** In the **Web Service Extensions** panel (in the right frame), click **Add a new Web service extension...**.
- **c.** In the **New Web Service Extension** dialog, enter `IISProxy` in the **Extension name** field.
- **d.** Click **Add...**.
- **e.** In the resulting dialog, navigate to and select the `iisproxy.dll` file you copied to `${iis_home}\scripts` in step 2.
- **f.** In the **New Web Service Extension** dialog, select the **Set extension status to Allowed** checkbox.
- **g.** Click **OK**.
- **h.** Restart IIS.

20. Select and right-click **Default Web Site**.

21. In the resulting menu, select **Properties**.

22. In the **HTTPS Web Site Properties** dialog, open the **ISAPI Filters** tab.

A green, up-pointing arrow displays in the **Status** column:



Filters installed here are active for this Web site only. Filters are executed in the order listed below:

Status	Filter Name	Priority
↑	IISForward	High

If the **Priority** column displays **Unknown**, access the Web site (for example, `http://server.domain.com`) in a Web browser. This will cause IIS to load the filter.

23. Close IIS Manager.

7.3 Setting up and using WebSuite

7.3.1 Importing static data from TRM

Once the installation of WebSuite is complete, the administrator must import static data from TRM and configure it according to the organization's requirements. These procedures are explained in the *WebSuite User Guide*.

The *WebSuite User Guide* also describes the business functions of WebSuite: trading, generating reports, monitoring the treasury position and entering a cashflow forecast.

7.3.2 Configuring report layouts

Note: There is a known error with reports when running TRMWeb as a seamless application on a Citrix Presentation Server version 4.5. However, under Version 4.0 there is no such problem.

To configure the report layouts, copy the files from:

<install dir>`/components/wss-web/trmweb/builtin/conf/fk/share/reports/layouts/trmWeb`

to the `site` directory:

<site dir>`/base/components/trm/share/reports/layouts/trmWeb`

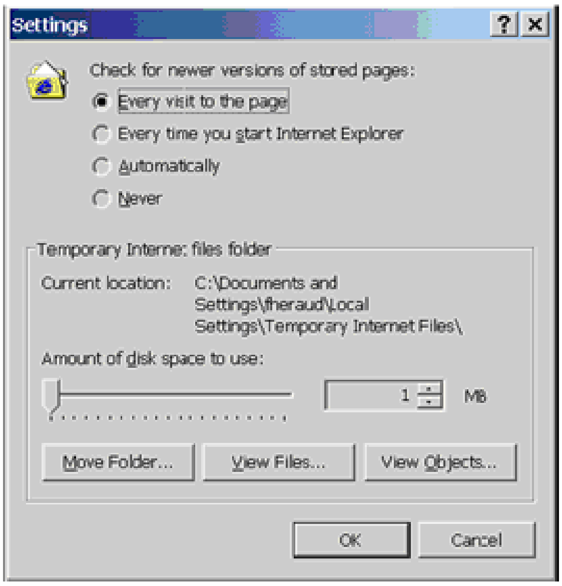
and execute a `site` command.

7.3.3 Configuring the Web browser for WebSuite

Define the Web browser parameters (for example, to check for newer versions of stored pages on every visit, enable JavaScript, and so on).

Note: You can only use WebSuite with Internet Explorer 5.0, 5.5, and 6.0.

1. Launch Internet Explorer.
2. Select **Tools - Internet Options**.
3. Select the **General** tab.
4. In the Temporary Internet files section of the page, click **Settings**.



Check for newer versions of stored pages:

● Every visit to the page
○ Every time you start Internet Explorer
○ Automatically
○ Never

Temporary Internet files folder

Current location: C:\Documents and Settings\fheraud\Local Settings\Temporary Internet Files\

Amount of disk space to use: 1 MB

5. To check for newer versions of stored pages, select the option **Every visit to the page**.
6. Click **OK**. Temporary internet files will now be refreshed each time they are requested.
7. Select the **Private** tab.
8. Click **Advanced...**.
9. Select the following options:
   - **Override automatic cookie handling**
   - **First-party cookies**: `Accept`
   - **Third-party cookies**: `Accept`.
10. Click **OK**. Cookies are now enabled on your Web browser.
11. Select the **Advanced** tab.

7.3.3.1 Starting WebSuite

After logging onto WebSuite, there are navigation menus available for TRM, CMM, and ACM depending on what was selected at installation time.

7.3.4 WebSuite and resource files

When WebSuite is launched, it looks for resource files in the folders represented by the following environment variables:

1. `%WEBSUITE_ENV_HOME%\resources`
   `(\ws-suite\envs\wss73\etc\wss-web\resources)`

   If this folder does not exist, then it looks in:

2. `%WEBSUITE_ENV_CONF_HOME%\resources`
   `(\ws-suite\envs\wss73\etc\wss-web\websuite\resources)`

If this folder does not exist, it looks in:

3. `%WEBSUITE_CONF_HOME%\resources`
   `(\ws-suite\components\wss-web\websuite\resources)`

   When you install WebSuite, it creates this folder and copies its resource files here. When WebSuite is upgraded, new versions of the resource files replace whatever is already in this folder.

This means that you can make two sets of changes to the resource files and deploy them as you wish by using this prioritized folder structure (but you must first create the folders in points 1 and 2 above). You can also protect them from being overwritten by an upgrade.

7.4 TRMSwift configuration changes since the last version

TRMSwift is delivered with a set of HTML files that summarize the changes made in its XML configuration since the previous version. These files are located in the `$FK_HOME/etc/trmswift/package/package.changes` directory. This directory also contains `trmswift.config.txt` which is a flat representation of the entire TRMSwift configuration.

Chapter 8 Upgrading: Wallstreet Suite

This chapter describes how to upgrade Wallstreet Suite to the latest version.

8.1 Pre-installation process

Before an upgrade, you should perform the following tasks:

1. Download the new package.

   Connect to our UIS web site and download the requested version.

2. Shut down running processes.

   Stop all Wallstreet Suite processes that are running. Normally, you use Process Monitor to stop all processes, then manually stop the **pmsd** process and all **pmad** processes.

3. Dump the database.

   We advise you to do a dump of your database as a safeguard against any errors during the upgrade procedure.

8.2 Installing the new version

Follow the instructions in *5.2 First-time installation* on page 57.

Important: If the previous installation had a `site` directory you must re-use it.

The content of this directory must be compared and validated with the changes that come from the new version. Once done, you can apply changes using the reconfigure procedure.

8.3 Database upgrade

Important: If you want to create and manage an environment without SDM (**not recommended with versions higher than Wallstreet Suite 7.3**), launch the SIWizard with the option `--no-sdm`.

8.3.1 Upgrading from a version before 7.1.0.1

If you have a version of Wallstreet Suite prior to version 7.1.0.1, see *Chapter 9 Upgrading: database upgrade to Wallstreet Suite version 7.1.0.1* on page 101. When you have a 7.1.0.1 version, continue with the rest of this chapter.

8.3.2 Upgrading from version 7.1.0.1 onwards

There is a single script to run which performs the upgrade from version 7.1.0.1 onwards to the latest version of 7.3 and beyond.

8.3.2.1 Verifying version 7.1.0.1

If you are upgrading from version 7.1.0.1, verify that your TRM database is set to version 7.1.0.1:

```
select value from Configuration where id = 10
```

If you are sure that you have a 7.1.0.1 database, but the Configuration table value is not 7.1.0.1, change it manually:

```
update Configuration
  set value = '7.1.0.1'
  where id = 10
go
```

8.3.2.2 Preparing for upgrade

Before performing the upgrade, follow these steps:

1. Back up the database that you are going to upgrade.
2. Open the file `..sharedconf\components\database\upgrade.properties`, and make the following changes:
   - If TRMWeb is already installed, set `upgrade.trmweb.already.installed=True`
   - If TRMSwift (iKIT) is already installed, set `upgrade.trmswift.already.installed=True`
   - If TRMSwift (iKIT) is already installed and:

     if the iKIT table prefix in the database is already `IKIT`, set `upgrade.trmswift.table.old.prefix=NA`

     if the iKIT table prefix in the database is not IKIT, set `upgrade.trmswift.table.old.prefix=`*`prefix`* where *`prefix`* is the actual prefix.
   - If TRMSwift (iKIT) is already installed and:

     if the iKIT archive table prefix in the database is already `IKAR`, set `upgrade.trmswift.table.archive.old.prefix=NA`

     if the iKIT archive table prefix in the database is not `IKAR`, set `upgrade.trmswift.table.archive.old.prefix=`*`prefix`* where *`prefix`* is the actual prefix.

8.3.2.3 Upgrading CMM users (CMM with TRM)

If CMM users exist that TRM does not know about, then you must manually create them using the Security Center application, and assign them to a web user group.

Note: The CMM part of the upgrade process creates backups of the `USERPROFILES` and `EMPLOYEE` tables (`USERPROFILES_TEMP` and `EMPLOYEES_TEMP`). Also, the Security Profiles table `OBJECTACLS` is kept.

8.3.2.4 Upgrading the database using the Suite Installer Wizard (recommended)

Important: To use this procedure, the existing database must have SDM already enabled, otherwise you should use the procedure in *8.3.2.5 Upgrading the database using the Suite Installer command line* on page 91 instead.

Follow these steps:

1. Start up the Suite Installer Wizard as described in step *2.* on page 58.
2. On the Type of installation page, select **Database Management**.
3. On the next page, select **Database Upgrade**.
4. On the next page:
   - Check the paths are correct for the installation and `site` directories.
   - Select the correct environment from the drop-down list.
5. The Suite Installer Wizard displays a dialog box requesting the database SYS or sa password (depending on your database type), so that it can run queries on the installed database.
6. The next screen requires you to click **Upgrade** to start the process.

The Suite Installer Wizard checks that the Wallstreet Suite components are at the correct version.

CMM upgrade is done on all databases, including Sybase where WebSuite (CMM) is not available. This is because CMM database objects are required by TRM, even when CMM is not installed.

Upgrade scripts are dynamically generated by the Suite Installer. The content of these scripts is displayed to the user before execution, and can be found here:
`SI_Tools/SI_Wizard/logs/`*`DD_MM_YY_n`*.

For example: `upgradeTRM_1.bat` and `upgradeACM_1.bat`.

8.3.2.5 Upgrading the database using the Suite Installer command line

Note: If the existing database already has SDM enabled, we recommend that you upgrade the database using the procedure in *8.3.2.4 Upgrading the database using the Suite Installer Wizard (recommended)* on page 91.

You can generate upgrade scripts using Suite Installer GUI and cancel their execution. This makes upgrade scripts available for a command line upgrade.

8.4 Instrument and Transaction Migration (pre 7.1.01 only)

Note: If you are upgrading from version 7.1.0.1 or above, ignore this section.

When upgrading from version 6.5 to 7.3, it is necessary to migrate the Instruments and Transactions to the new framework.

TRM 7.3 covers more instruments than did previous versions. To simplify instrument setup and to allow for expansion, the underlying instrument data model has been changed. Changes include removal of the market, feature-oriented architecture, removal of most of the switches and properties, support for parallel accounting and introduction of instrument groups.

8.4.1 Operations

8.4.1.1 Upgrade

In a regular TRM upgrade, you run the database upgrade scripts delivered with the package and to install the new binaries, as described in the previous sections. These upgrade scripts are SQL batch files which change the data model and the system data to align the database structure with the target version. After the upgrade scripts have been run, the database stored procedures must be rebuilt.

8.4.1.2 Instrument migration

The upgrade scripts prepare the new TRM data structures for instruments, but do not move the instrument data from the old structures to the new ones. The instrument migration tool must be used after database upgrade and rebuild of all the objects. It copies most instruments and related information from the old to the new data structures. Details of instrument migration tools are included here.

Some manual setup is required after instrument migration, as some instruments have been reworked so much that it would be inefficient to try to migrate them automatically.

The key entities to be set up, or at least completed manually, include:

- All the entities related to result configuration
- Portfolios and clients with respect to result configuration
- Currencies valuation configuration
- Volatility surfaces
- Instruments out of migration scope
- Facilities and tranches for commercial lending

8.4.1.3 Transaction Migration

Data structure changes are applied to the transactions and cashflow tables, and new data structures related to transactions are created, but the business-related part of the job is not done by upgrade scripts. Transaction migration is similar to instrument migration and it is detailed in this document. This tool must be used AFTER the steps above have been completed.

As in the case of instruments, some transactions are not handled by the migration tool, so must be re-entered

8.4.2 Migrating instruments

Instrument migration can be run through a TRM report (Migrate Instruments). You can define the instrument coverage and when you run the report, the instruments are automatically migrated. The tool scans all the instruments defined for coverage (see parameters below for details) and decides how they will be moved to the new data model.

A detailed log is generated and stored in the database for every execution of the tool.

Note: This tool does not update the old instruments. It means that it is possible (for testing purposes) to run the tool on the same data as often as needed without having to backup/restore data on each iteration.

8.4.2.1 Instrument type

In TRM 7.3, Instrument Types are used in instrument configuration. Instrument Types from 6.5 are not used when converting to 7.3 Instrument Types (but they are used for Instrument Groups, see below). When converting instruments from 6.5, there is no need for extensive usage of Instrument Types, since the instruments are already configured. For this reason, only one Instrument Type per Instrument Class (with all available features) is created by the migration tool.

If you want to set up custom types in parallel, you must ensure that none of the IDs are the same as a class ID.

8.4.2.2 Instrument group

TRM 7.3 uses hierarchical Instrument Groups. This replaces Market and Instrument Types, except in instrument configuration described above. In the migration process, the new instrument groups will replace the old instruments' Market (level 1) and Instrument Type (level 2).

The migration tool creates the groups and then assigns the correct group to the migrated instruments to replace the old Market and Instrument Type.

8.4.2.3 Other data

Other instrument-related data like user-defined properties, prices, market information are automatically attached to the migrated instruments with the appropriate modifications.

8.4.2.4 List of instruments covered

Instrument	Market	Instrument Migration	Transaction Migration
Short-loans & deposits	MM	Yes	Yes
Discount papers	MM	Yes	Yes
Fixed & floating rate loans	MM	Yes	Yes
Vanilla Bonds (including FRN)	MM	Yes	Yes
Vanilla IR swaps (including cross currency but excluding OIS)	MM	Yes	Yes
FX-Spots & forwards	FX	Yes	Yes
FX swaps	FX	Yes	Yes
Vanilla OTC FX options	FX	Yes	Yes
Barrier FX Options	FX	Yes	No
Equity shares	EQ	Yes	Yes
Equity options	EQ	Yes	Yes
Funds	EQ	Yes	Yes
Balances	BA	Yes	Yes
Call Money/Account	MM	Yes	Yes
Cap floors & collars (without barriers)	MM	Yes	Yes

Vanilla swaptions (without barriers)	MM	Yes	Yes
IR Futures	MM	Yes	Yes
Payments & Transfers	BA	Yes	Yes
Facility - Drawdowns	MM	Yes	Yes
Repo/Buy Sell Back	MM	Yes	Yes

8.4.2.5 Parameters

For instrument migration, the following parameters are available:

- Instrument: The id of the instrument to migrate. If this parameter is used, only this instrument will be migrated regardless of the following parameters.
- Type: Migrate only the instruments belonging to this type.
- Market: Migrate only the instruments belonging to this market.
- Domain: Migrate only instruments belonging to this domain. If none of the above parameters are filled, the whole database will be migrated.
- Reset Flags: This flag is to be used only when you want to re-migrate an instrument which has already been migrated. In this case when the flag is not set it will try to keep as many manual setup as possible (it is to be used when you have already configured some result, valuation or new features on the instrument that you want to keep when trying a new version of the migration tool). If the instrument has never been migrated, it has no impact. If you want to reset result configuration use 1, for valuation configuration use 2, for configuration of new features use 4, and any combination of those (1+2+4=7 resets everything).
- Get Log: When this flag is on the log of the execution is displayed, it is always possible to see execution logs afterwards using the Migration Log report.

8.4.2.6 Tuning

The following properties and tables can be used to tune the migration. They must be created by PS in the version 6.5 Property Editor and assigned to corresponding instruments using 6.5 instrument editors.

As a workaround, these properties can also be created in the migrated 7.3 database using SQL inserts. See below example that creates the property and assigns it to an instrument.

For example:

```
insert into Property (object_id, type_id, value_id, name)
  values ('Instrument', 'MIGRATE-INSTRUMENT-ID', 1, 'MIGRATE-INSTRUMENT-ID')
```

Same for object_id: FXInstrument, Equity and BankEventType

Then, the script would assign the property automatically to the instruments:

For example:

```
insert into PropertyMap (object_id, key_id, type_id, value)
  values ('Instrument', '<OLD-INST-ID>', 'MIGRATE-INSTRUMENT-ID', '<NEW-INST-ID>')
```

Same for object_id: FXInstrument, Equity and BankEventType

8.4.2.6.1 MIGRATE-INSTRUMENT-ID property

The instrument migration tool does a one-to one conversion (one "old" instrument generates one "new" instrument **with the same ID**). To avoid this (reduce the number of instruments, for example) add the property MIGRATE-INSTRUMENT-ID to the old instrument, with the id of the new instrument.

All the instruments having this property will be ignored by the migration and the new target instruments will have to be created manually (unless the target is an instrument which is automatically converted).

Note: To exclude an instrument from the automated migration, use a property with a null value.

8.4.2.6.2 MIGRATE-CLASS-ID property

The instrument migration tool tries to detect the 7.3 instrument class to which it will belong. In some cases, it is not possible to detect it consistently (due to characteristics of the 6.5 frameworks, or non standard setup).

Instead, you can specify how the instrument will be migrated by setting the property “MIGRATE-CLASS-ID” on the instrument in 6.5 instrument editors (Equity Editor, FX Instrument Editor, IR Instrument Editor, Payment Instrument Editor). When it is set, the migration tool will automatically assume the instrument has to belong to this class.

For example, IR swap legs were swap instruments in 6.5 (switch Swap is ON in IR Instrument Editor), but those are loan instruments in 7.3. To have 6.5 swap leg instruments migrated correctly, use property MIGRATE-CLASS-ID with class LOAN

8.4.2.6.3 MigrateByType table

This tuning option is similar to the previous one, but at the old type level. A table called MigrateByType has been added to the data model. It contains the columns market_id, type_id, migrate_class_id.

If you fill in a row, instruments belonging to this market and type must be mapped to the specified 7.3 class. The table must be filled directly using an SQL tool.

8.4.2.6.4 MigrateByClass table

This is similar to MigrateByType, but at the old class level. It applies only to money market instruments, as the concept of class does not exist for other markets in version 6.5

A table MigrateByClass has been added to the data model. It contains the columns market_id, class_id, migrate_class_id.

If you fill in a row, it means that instruments belonging to this market. Class in ‘6.5- version’ must be mapped to the specified 7.3 migrate_class_id. The table must be filled directly using an SQL tool.

8.4.2.6.5 MIGRATE-RESULT property

The setup of results is new in 7.3: the corresponding functionality did not exist in 6.5. You must set up the result manually using Result Type Editor and Result Editor. Use the property MIGRATE-RESULT to specify the result type for a migrated instrument. When no type is defined, the result type DEFAULT is used.

8.4.2.6.6 MIGRATE-BOND-CONVENTION property

The bond convention is detected from the 6.5 setup, but the process may not work reliably for some instruments such as government bonds. You can force a convention by using the property MIGRATE-BOND-CONVENTION. A list of available conventions can be found in Instrument Editor, Trading Yield page, Yield Convention field. For the mapping to work, the Value property must match the intended Yield Convention ID.

8.4.2.6.7 MIGRATE-UNDERLYING property

This property is used to specify the FX spot/forward instrument that should be used as underlying instrument for FX options. The specified value is used to set up the FX option instrument, FX Option page, field **Exercise Instrument**.

8.4.2.6.8 MIGRATE-SCHEDULE-TEMPLATE-ID property

This property is used to migrate long-term loan instruments and bond instruments. The standard migration creates one schedule template for each loan instrument. This template is then assigned to the instrument, **Generic Loan** page, field Structure. When this property is assigned, its value is used instead of the default.

For bond instruments, standard migration uses templates FIXED-BULLET, FLOATING-BULLET, ZERO-COUPON. When the property is specified, the corresponding schedule is used as the basis for bond schedule.

8.4.2.6.9 MIGRATE-DD-INSTRUMENT-ID property

This property is used to specify the instrument that should be used when migrating CLM related transactions. In 7.3, the new instrument class COMMERCIAL-LOAN exists. This property must be used when specifying a new instrument to migrate CLM related transactions,

8.4.2.6.10 MIGRATE-COLLATERAL-AGREEMENT property

There is new architecture in place for Repo functionality. Consequently, it is necessary to specify the collateral agreement that should be used for given repo instruments. You must create the agreement manually in Collateral Agreement Editor and assign it to the instrument by using this property.

8.4.2.6.11 MIGRATE-SUBSTITUTION-INSTRUMENT-ID property

Substitution transactions were previously using repo instrument in version 6.5. Now they have to use instrument on class SUBSTITUTION. There are two possibilities to define which instrument to be used on migrated substitution transactions:

1. Define substitution instrument using property MIGRATE-SUBSTITUTION-INSTRUMENT-ID
2. Define default substitution instrument for given collateral agreement using Collateral Agreement Editor

8.4.2.6.12 MIGRATE-MARGIN-INSTRUMENT-ID property

Margin transactions use instrument class MARGIN-MOVEMENT in 7.3. To get correct mapping, assign this property to the instrument, which was used in 6.5. Default value could be coming from collateral agreement setup.

8.4.2.6.13 MIGRATE-FEATURE-* property

To force the addition or removal of a feature during migration, create a property for a feature, with prefix MIGRATE-FEATURE- and type Boolean. When property is defined for a given 6.5 instrument with value Yes and feature exists for corresponding instrument class, the feature is added to the migrated 7.3 instrument. When the value is No, the feature is removed.

For example, to replace the feature FIXED-IR-METHOD with FX-METHOD for your cash instrument, add the property MIGRATE-FEATURE-FIXED-IR-METHOD with value No and property MIGRATE-FEATURE-FX-METHOD with value Yes to your 6.5 instrument. During migration, the default feature FIXED-IR-METHOD will be removed and the specific feature FX-METHOD will be added.

8.4.2.7 Reporting

The migration tool populates the following database tables for reporting purposes: MigrateLog, MigrateLogObject and MigrateLogMessage.

A row is inserted in the migration log for every execution of the migration tool. Then, referencing the above, a row is inserted in MigrateLogObject for each migrated instrument. Finally, a message is inserted in MigrateLogMessage each time an abnormal case is detected or each time the tool detects that some manual complementary setup will have to be done after migration. Messages can be errors, warnings or information (severity 0 to 2).

The Migrate Log report provides a log of previous executions. It is available from **Applications -**

Wallstreet Suite - Transaction and Risk Mgmt -- Administration Tools -- Migration Reports.

8.4.3 Migrating transactions

Transaction migration is done with a tool executed through the python script migrate.py, located in the folder share/python. When the script is executed, the transactions are automatically migrated. Transaction migration log is created and it can be displayed using Report Generator.

The tool scans all the transactions defined by the perimeter (see parameters below for details) and decides how it will be moved to the new data model. A detailed log is generated and stored in the database after every execution of the tool.

Note: The tool backs up the old transactions, cashflows etc before updating them. It means that it is possible (for testing purposes) to run the tool as many times as needed on the same data, without having to backup/restore data on each iteration.

If the tool detects that a transaction has already been migrated, it restores the original data from the backup tables before re-running the migration on original data. This backup/restore process can reduce the performance of the tool: to save time, it can be disabled when the final production migration is done

The migration tool keeps the transaction numbers and the cashflow ids.

8.4.3.1 Instrument coverage

The coverage for transaction migration is the same as for instrument migration. The transactions and all the related entities (cashflows etc) are migrated by the tool. Special transactions like average balance transactions and mirrored transactions are included.

8.4.3.2 Parameters

The following parameters are available for transaction migration:

p_number: The transaction to migrate (to test the tool for only one transaction)

p_instrument_id: If specified, only the transactions on this instrument will be migrated

p_type_id: If specified, only the transactions on this instrument type will be migrated

p_portfolio_id: If specified, only the transactions in this portfolio will be migrated

p_market_id: If filled, only the transactions belonging to this market will be migrated

p_closing_date_min: Allows setting a cut-off date: if filled, the transaction closed before this date will be ignored by the migration tool

p_closing_date_max: If a cut-off date has been used, this parameter allows you to migrate the history in a later stage – possibly, after live production in 7.3 has started

force_p: If the flag is set to 1, the transaction will be re-migrated. Having force_p set to 0 or leaving it blank will prevent multiple migration of transaction.

verbose_p: If the flag is set to 1, the log will be displayed after migration is completed.

backup_p: If the flag is set to 1, the transactions are backed up.

inventory_p: If the flag is set to 1, the inventory is updated.

notification_p: If the flag is set to 1, the transaction is notified, so real-time message is propagated within the system.

accounting_p: If the flag is set to 1, the accounting inputs are created

8.4.3.3 Tuning

As described above, the Migrate Instrument ID property has an impact on transaction migration. The transaction migration tool uses the class of the new instrument to determine how the transaction is to be migrated, so the other instrument tuning options have an indirect impact on transaction migration.

The property FACILITY-ID allows you to map a facility/tranche to a drawdown. The facility must be created manually in Facility Editor.

8.4.3.4 Reporting

Reporting is similar to reporting for instruments except that the objects are transactions. The Migrate Log Report handles transactions and instruments at the same time.

8.4.3.5 Transaction migration script

The parameters of the elementary procedure used for technically migrating transactions are described in section 4.2. However, additional tasks must be performed afterwards to complete the migration of transactions.

The migration.py script located in $FK_HOME/share/python executes the following steps:

- Migrates transactions and return a list of transactions migrated
- For each transaction:
  - Modify transaction and cashflows to fit new data model, create other corresponding entities (schedules, legs …)
  - Creates Accounting Input
  - Updates the inventory (populates internal technical information)

Note that assigning classifications and triggering parallel result treatments for existing transactions is out of the normal migration scope.

The transaction migration script can help the customer to start to take advantage of the new functionality for old positions, but it is a complex process and there are probably special cases which are not described here.

8.5 Migration of cost-of-carry transactions

If you are already in a version 7.1+, you can skip this chapter.

When upgrading from version 6.5 to 7.3, it is necessary to migrate the cost-of-carry transactions to the new framework.

Following the restructuring of Cost of Carry functionality, extra steps have to be taken into account during migration.

In 6.5, there was one transaction for each day. Such transactions contained one balance flow and one interest flow for each currency. The instrument used was hardcoded as COST-OF-CARRY.

In 7.3, there is one transaction per currency. Such transactions contain one interest flow for each day and one balance flow for each date when balance amount was changed. Instruments derived from class COST-OF-CARRY are used.

To port instruments and transactions to the 7.3 framework:

1. New instruments are generated according to data (rate to be used etc.), defined in 6.5 for portfolios and activities. Instrument is the same as portfolio id with prefix COC-. This is part of the standard upgrade, no manual action required.
2. After a full upgrade, new cost-of-carry transactions are generated by running the procedure MigrateCoCTransaction once. This procedure creates new cost-of-carry transactions and copies cashflows from transactions.
3. If you want to use other than the default values for instrument, define them in portfolio property MIGRATE-COC-INSTRUMENT-ID.

8.6 Activating the new yield curve setup

Important: These steps must be run only after the end of the upgrade from version 7.2 or older to version 7.3.

Note: Suite Installer version 7.3 onwards performs these steps automatically if you use its upgrade feature.

After Suite Installer has finished upgrading the database to 7.3, the following steps activate the new Yield Curve setup in SDM:

1. Run this SQL script:

   ```
   $FK_HOME/share/<db-platform>/upgrade/IR_SDM_Upgrade_73.sql
   ```

   This command drops all T-prefixed tables of static data entity IR.

   Run the following command:

   ```
   perl build.pl -v -k -E <path-to-this-directory>/IR_SDM_Upgrade_73.sql > log1.log
   ```

   If the log file contains an error stating that the table `TIRCDSwap` could not be dropped because it did not exist, ignore it.
2. Run the upgrade again.

   This command should do almost nothing apart from correctly populating T-prefixed tables of static data entity IR.

   Use the following command:

   ```
   perl build.pl -X -v -w > log2.log
   ```

8.7 Upgrading CMM installation data

If CMM was previously installed, follow these steps:

1. In the original CMM Application Server hierarchy, locate the `InstallationData` folder, and copy its contents (not the folder itself) to:

   ```
   \envs\<env>\wss-web\ConfigurationData
   ```
2. Merge the contents of the old and new files called `nvp.xml`:

a. Open the old file (that you have just copied):

   `\envs\<env>\etc\wss-web\cmm\ConfigurationData\installation\nvp.xml`

b. Open the new file which the upgrade process created for you:

   `\envs\<env>\etc\wss-web\cmm\InstallationData\installation\nvp.xml`

c. Merge the two and save the file to the location in step *b.* above. (The old file can be deleted.)

8.8 Upgrade the TRMSwift configuration

Use this procedure if your system uses TRMSwift.

1. Navigate to this folder `ws-suite/components/trm/etc/trmswift/package`
2. Merge your Customer Specific Developments (CSDs) into these XML files.
3. Load the TRMSwift configuration into the database:
   a. Start the shell that corresponds to your environment. For example, if your environment is called dev, then start:

      Windows:

      ```
      C:\WSS\ws-suite\envs\dev\bin\trm-dev-shell.bat
      ```

      Unix:

      ```
      /wss/ws-suite/envs/dev/bin/trm-dev-shell.sh
      ```

   b. Run this command in the shell: `trmswift update`

Note:

Chapter 9 Upgrading: database upgrade to Wallstreet Suite version 7.1.0.1

This section describes how to upgrade older versions of Wallstreet Suite. It covers upgrades from from Finance KIT 6.5 onwards. This should be performed by a skilled team, and only by prior agreement with WSS.

You will be using the WSS-provided script build.pl. For full information about this script, see *Appendix D The build script* on page 139.

Note: The "…" characters in the examples refer to the standard user name, password, and database name.

Important: If your upgrade includes enabling Static Data Manager (SDM), you should complete all upgrade steps below before enabling SDM. In other words, you should have an upgraded and fully-functional Wallstreet Suite, and then follow the procedure in *Appendix C Enabling SDM on a non-SDM system* on page 135.

CMM and ACM

If your installation already includes CMM or ACM, there are database upgrades to be performed after those described in this section. See the relevant Installation Guides for CMM and ACM.

9.1 Oracle upgrade

9.1.1 Prerequisites

9.1.1.1 Sybase emulation library

Before running any upgrade scripts, upgrade the Sybase emulation library to the latest version by running this script:

```
cd $FK_HOME/share/oracle
build -U … -P … -D … -v -k -p syblayer1
```

9.1.1.2 sqlplus and UTF8 encoding

All SQL scripts use UTF-8 encoding. Therefore it is important that you set the environment variable NLS_LANG correctly: this variable is the only way to tell the Oracle tool sqlplus how it should interpret characters in the upgrade scripts.

For details on setting NLS_LANG, see the *Oracle9i Database Globalization Support Guide (Release 2 (9.2))*.

For any SQL scripts to run correctly, the charset part of the NLS_LANG variable must be set to UTF8. Using the default values for language and territory, set NLS_LANG like this:

```
set NLS_LANG=_.UTF8
```

9.1.1.3 Existing TRMSwift (Integration KIT) users

If your existing installation already has TRMSwift/Integration KIT installed, then check which version you are running.

If you have Integration KIT (iKIT) version 6.5.5.0 or older, then upgrade this to the latest 6.5.x.x version of iKIT available on the WSS UIS site (`http://www.trema.com/support`). See *Chapter 10 Upgrading: iKIT* on page 109.

If you have a version of Integration KIT (iKIT) later than 6.5.5.0, then continue reading.

9.1.2 Oracle: upgrade steps

Note: All `build` commands must be run in the `$FK_HOME/share/oracle` directory.

You should search the log files that result from `build` commands for ORA- and SP2- substrings to ensure that there were no errors during the upgrade.

9.1.2.1 Preparing to upgrade

1. Shut down all your real-time daemons. Make sure that you do not have the client software running. Below is an example only:

```
ps -eaf | grep fk6.5 | grep -v grep
kill -9 <pid> ....
```

2. Download and unzip the latest Wallstreet Suite package from the Wallstreet Systems Support site.

3. **Important:** before starting the upgrade, make a dump of the database.

4. Depending on which version of Wallstreet Suite (or Finance KIT) you are upgrading, choose the correct starting point in the rest of this section.

9.1.2.2 Upgrading from Finance KIT 6.5 to Wallstreet Suite 7.1.0.0

Upgrade scripts are supplied in this version of Wallstreet Suite for upgrading Finance KIT 6.5.7.2 or earlier. If you have Finance KIT 6.5.7.3 or later, contact the WSS R&D department.

To perform an upgrade, run the SQL scripts in the `$FK_HOME/share/oracle/upgrade/7.1` directory in alphabetical order, using this command:

```
build -U … -P … -D … -v -e -E upgrade/7.1/\*.sql > upgrade.71.log 2>&1
```

You have now upgraded the database to Wallstreet Suite version 7.1.0.0.

9.1.2.3 Upgrading from Wallstreet Suite 7.1.0.0 to 7.1.0.1

From the `$FK_HOME/share/oracle` directory, run the following `build` command:

```
build -U … -P … -D … -v -e -E upgrade/7.1/7.1.0.1/\*.sql > upgrade.7101.log 2>&1
```

You have now upgraded the database to Wallstreet Suite version 7.1.0.1. Now go to *8.3.2 Upgrading from version 7.1.0.1 onwards* on page 90.

9.2 MSSQL upgrade

9.2.1 Prerequisites

9.2.1.1 Existing TRMSwift (Integration KIT) users

If your existing installation already has TRMSwift/Integration KIT installed, then check which version you are running.

If you have Integration KIT (iKIT) version 6.5.5.0 or older, then upgrade this to the latest 6.5.x.x version of iKIT available on the WSS UIS site (`http://www.trema.com/support`). See *Chapter 10 Upgrading: iKIT* on page 109.

If you have a version of Integration KIT (iKIT) later than 6.5.5.0, then continue reading.

9.2.2 MSSQL: upgrade steps

Note: All `build` commands must be run in the `%FK_HOME%/share/mssql` directory.

You should search the log files that result from `build` commands to ensure that there were no errors during the upgrade.

9.2.2.1 Preparing to upgrade

1. Shut down all your TRM server processes. Make sure that the client software is not running.
2. Download and unzip the latest Wallstreet Suite package from the Wallstreet Systems Support site.
3. **Important:** before starting the upgrade, make a dump of the database.
4. Make sure that MSSQL isql tool is available in your `PATH`, for example:

```
set PATH="C:\Program Files\Microsoft SQL Server\80\Tools\Binn";%PATH%
```

5. Set your TRM environment so that you can use TRM environment variables:
   a. Open a Windows Command Prompt.
   b. Change to the `%FK_HOME%\bin` directory, for example:
      `C:\TRM\fk\bin`
   c. Run fk `MSSQL-database-name_sql`, for example:
      `C:\TRM\fk\bin>fk WSStest01_sql`
   d. When the login screen appears, click the **Cancel** button and ignore any error messages.

9.2.2.2 Unicode

If you are upgrading from Finance KIT 6.5 to TRM 7.1, then all database `varchar` fields must be converted to `nvarchar` fields. This must be done before following the upgrade steps below:

1. Change to this directory:

```
cd %FK_HOME%\share\mssql\upgrade
```

Note: The next two steps can take a long time to complete.

2. Run this script:

```
perl unicode.pl -U fk -P fkfkfk -S %FK_DB_SERVER% -D %FK_DATABASE% -o
unicode_convert.sql
```

This creates an sql script that:

- Renames all `varchar` user datatypes.
- Creates all renamed datatypes as `nvarchar`.
- Drops all indexes on `varchar` columns.
- Alters all `varchar` columns in all tables to `nvarchar`.
- Drops and recreates all procedures and views which use renamed user datatypes.
- Drops renamed user datatypes.

3. Submit this script to the database:

```
type unicode_convert.sql | isql -U fk -P fkfkfk -S <full name of database server>
-d %FK_DATABASE% -o database_unicode.log
```

4. Look in the output file `database_unicode.log`. You can safely ignore any warnings that some procedures were created with problems because they depend on missing objects - provided that the 'missing object' is one of these procedures:

   - `AccountUpdateBalance`
   - `GetFundReportHeaderTmp`
   - `ReadKeyFigureNumbers`
   - `ProcessClosingTheBooksPeriodic`
   - `ReadCreditRiskNumbers`

   Warnings have this form:

```
Msg 2007, Level 11, State 1:

Server 'MSSPROD1', Procedure 'CalculateBalances', Line 182:

Cannot add rows to sysdepends for the current stored procedure because it
depends on the missing object 'AccountUpdateBalance'. The stored procedure
will still be created.
```

5. The script `unicode_convert.sql` drops all indexes on columns of type `varchar`. So that subsequent upgrade steps run optimally, at this point you should rebuild all indexes in the database.

   a. Open a new shell and evaluate in it the environment of your old TRM installation (i.e. version 6.5).

   b. `cd %FK_HOME%\share\mssql`

   c. `perl build.pl -U ... -P ... -S ... -D ... -v -i all`

9.2.2.3 Upgrading from Finance KIT 6.5 to Wallstreet Suite 7.1.0.0

Upgrade scripts are supplied in this version of Wallstreet Suite for upgrading Finance KIT 6.5.7.2 or earlier. If you have Finance KIT 6.5.7.3 or later, contact the Trema R&D department.

To perform an upgrade, run the SQL scripts in the `%FK_HOME%\share\sybase\upgrade/7.1` directory in alphabetical order, using this command:

```
perl build.pl -U fk -P fkfkfk -S %FK_DB_SERVER% -D %FK_DATABASE% -v -k -E
upgrade\7.1\*.sql > P7100.log 2>&1
```

You have now upgraded the database to Wallstreet Suite version 7.1.0.0.

9.2.2.4 Upgrading from Wallstreet Suite version 7.1.0.0 to 7.1.0.1

From the `%FK_HOME%\share\mssql` directory, run the following `build` command:

```
perl build.pl -U … -P … -S … -D … -v -k -E upgrade\7.1\7.1.0.1\*.sql > P7101.log 2>&1
```

You have now upgraded the database to Wallstreet Suite version 7.1.0.1. Now go to *8.3.2 Upgrading from version 7.1.0.1 onwards* on page 90.

9.3 Sybase upgrade

9.3.1 Prerequisites

9.3.1.1 Converting charset to UTF8

The following steps install Unicode support, and make it the default character set.

1. Install the character set into the server (Sybase 12 and 15 versions shown below):

   ```
   # $SYBASE/ASE-12_5/bin/charset <login opts> binary.srt utf8

   # $SYBASE/ASE-15_0/bin/charset <login opts> binary.srt utf8
   ```

2. Enable unicode conversions in the database:

   ```
   > sp_configure "enable unicode conversions", 1
   > go
   ```

3. Make Unicode the default character set:

   ```
   > sp_configure "default sortorder id", 50, "utf8"
   > go
   ```

4. Restart the ASE server. This will take time because it converts all the databases into utf8. Once the server finishes, it shuts down automatically, and needs to be restarted manually.

9.3.1.2 Existing TRMSwift (Integration KIT) users

If your existing installation already has TRMSwift/Integration KIT installed, then check which version you are running.

If you have Integration KIT (iKIT) version 6.5.5.0 or older, then upgrade this to the latest 6.5.x.x version of iKIT available on the Trema UIS site (`http://www.trema.com/support`). See *Chapter 10 Upgrading: iKIT* on page 109.

If you have a version of Integration KIT (iKIT) later than 6.5.5.0, then continue reading.

9.3.2 Sybase: upgrade steps

Note: All `build` commands must be run in the `$FK_HOME/share/sybase` directory.

You should search the log files that result from `build` commands to ensure that there were no errors during the upgrade.

9.3.2.1 Preparing to upgrade

1. Shut down all your real-time daemons. Make sure that you do not have the client software running.

   ```
   ps -eaf | grep fk6.5 | grep -v grep
   kill -9 <pid> ....
   ```

2. Download and unzip the latest Wallstreet Suite package from the Wallstreet Systems Support site.
3. **Important:** before starting the upgrade, make a dump of the database.

9.3.2.2 Unicode

If you are upgrading from Finance KIT 6.5, then all database `varchar` fields must be converted to `nvarchar` fields. This must be done before following the upgrade steps below:

1. Change to this directory:

   ```
   cd $FK_HOME/share/sybase/upgrade/
   ```

Note: The next two steps can take a long time to complete.

2. Run this script :

   ```
   perl unicode.pl -U … -P … -S … -D … -o unicode_convert.sql
   ```

   This creates an sql script that:

   - Renames all `varchar` user datatypes.
   - Creates all renamed datatypes as `nvarchar`.
   - Drops all indexes on `varchar` columns.
   - Alters all `varchar` columns in all tables to `nvarchar`.
   - Drops and recreates all procedures and views which use renamed user datatypes.
   - Drops renamed user datatypes.

3. Submit this script to the database:

   ```
   isql -U … -P … -S … -D … <unicode_convert.sql > unicode_convert.log 2>&1
   ```

4. Look in the output file `unicode_convert.log`. You can safely ignore any warnings that some procedures were created with problems because they depend on missing objects - provided that the 'missing object' is one of these procedures:

   - `AccountUpdateBalance`
   - `GetFundReportHeaderTmp`
   - `ReadKeyFigureNumbers`
   - `ProcessClosingTheBooksPeriodic`
   - `ReadCreditRiskNumbers`

   Warnings have this form:

   ```
   Msg 2007, Level 11, State 1:

   Server 'SYBPROD1', Procedure 'CalculateBalances', Line 182:

   Cannot add rows to sysdepends for the current stored procedure because it
   depends on the missing object 'AccountUpdateBalance'. The stored procedure
   will still be created.
   ```

5. The script `unicode_convert.sql` drops all indexes on columns of type `varchar`. In order to get decent performance of subsequent upgrade steps, it is recomended to rebuild at this point all indexes in the database.

   a. Open a new shell and evaluate in it the environment of your old TRM installation (i.e. version 6.5).

   b. `cd $FK_HOME/share/sybase`

   c. `build -U ... -P ... -S ... -D ... -v -i all`

9.3.2.3 Upgrading from Finance KIT 6.5 to Wallstreet Suite 7.1.0.0

Upgrade scripts are supplied in this version of Wallstreet Suite for upgrading Finance KIT 6.5.7.2 or earlier. If you have Finance KIT 6.5.7.3 or later, contact the Trema R&D department.

To perform an upgrade, run the SQL scripts in the `$FK_HOME/share/sybase/upgrade/7.1` directory in alphabetical order, using this command:

```
build -U … -P … -S … -D … -v -k -E "upgrade/7.1/\*.sql" > P7100.log 2>&1
```

You have now upgraded the database to Wallstreet Suite version 7.1.0.0. Continue the upgrade by following the steps below.

9.3.3 Upgrading from Wallstreet Suite version 7.1.0.0 to 7.1.0.1

From the `$FK_HOME/share/sybase` directory, run the following `build` command:

```
build -U … -P … -S … -D … -v -k -E "upgrade/7.1/7.1.0.1/\*.sql" > P7101.log 2>&1
```

You have now upgraded the database to Wallstreet Suite version 7.1.0.1. Now go to *8.3.2 Upgrading from version 7.1.0.1 onwards* on page 90.

Chapter 10 Upgrading: iKIT

This section describes how to upgrade from Ikit 6.x to IKit 6.5.5.0. For versions before 6.x, please contact our support.

10.1 Upgrading to the latest version of iKIT 6.5.x.x.

Follow these steps if your iKIT version is less than 6.5.5.0. This will generate some SQL files that you run to perform the correct update to the last 6.5 version.

1. Retrieve the parameters of your existing configuration.

Copy the `custom.properties` file that is provided with the 6.5 distribution (or the file that you use) into the directory `%FK_HOME%\etc\trmswift\upgrade`.

The file must be called `custom.properties`.

2. From the `$FK_HOME/etc/trmswift` directory, run the following command:

```
ant upgrade_to_last_65 -Ddbms_type=<sybase|oracle|mssql>
```

Example: `ant upgrade_to_last_65 -Ddbms_type=sybase`

This generates some SQL (6.5.4 to 6.5.5), and performs some checks. If you see an error similar to this:

```
[java] There is one error :  account is not defined in the SearchKeysMapping
files, please update this file before processing an update.
```

it means that you have some keys defined (maybe in a CSD) which are not part of the standard distribution/mapping setting. Modify the file `$FK_HOME/etc/trmswift/upgrade/xml/upgrade_searchkey.xml`, within `<SearchKeyMapping>`, and add a new key/mapping. Then rerun the command `ant upgrade_to_last_65 -Ddbms_type=<sybase|oracle|mssql>`.

Note: Whenever you see the error "...*reference<X>* is not defined...", you should add an equivalent entry (reference<X>) in the `setup.xml` file.

If you modify this file, be sure that your changes are back ported in a normal CSD file for `setup.xml` (in `$FK_HOME/etc/trmswift/package/GENERAL/setup/setup.xml`)

3. The following files will be generated in the `$FK_HOME/etc/trmswift/upgrade` directory:

```
6_1_0_0-to-6_1_1_1.sql

6_1_1_1-to-6_3_0_0.sql

6_3_0_0-to-6_5_0_0.sql

6_5_0_0-to-6_5_4_0.sql

6_5_4_0-to-6_5_5_0.sql 6_5_4_0-to-6_5_5_0_arch.sql
```

Run these files in order, starting with the filename that matches your version of iKIT.

10.2 Renaming the iKIT database tables

If your TRMSwift tables are not prefixed by IKIT (normal table) or IKAR (archive table), run this command:

```
ant rename -Ddbms_type=[sybase|mssql|oracle] -Dold.prefix=<old.prefix>
-Dold.archive.prefix=<old.archive.prefix>
```

where <old_prefix> and <old_archive_prefix> are the prefixes used in your previous installation.

Example: `ant rename -Ddbms_type=oracle -Dold.prefix=SH -Dold.archive.prefix=SHS`

The following files will be created in the `upgrade` directory which you should run to rename the tables:

```
rename_archivetables.sql
rename_tables.sql
```

10.3 Oracle only: moving TRMSwift tables to another schema

If you use another schema for TRMSwift tables, you can use the following target to generate scripts which will create and copy data from one schema to another.

```
ant move_schema -Dold.schema=<schema>
```

where `<schema>` should be replaced by the value of the old TRMSwift schema.

This creates `move_to_trm_schema.sql` in the `upgrade` directory. This script creates and copies all TRMSwift tables in the TRM schema.

Launch this script in the TRM schema with a user that has "read" rights to the old TRMSwift schema (to avoid permission errors).

10.4 Oracle only: cleaning the database

During the update process, some old features could be left in the system. Because of technical differences between standalone iKIT and TRMSwift, some of these are no longer needed.

To clean your database, launch the following command to generate the correct SQL:

```
ant.bat clean -Dold.prefix=<old.prefix> -Dold.archive.prefix=<old.archive.prefix>
```

Example: `ant.bat clean -Dold.prefix=SH -Dold.archive.prefix=SHS`

This creates `clean.sql` in the `upgrade` directory which you should run to clean the database.

Chapter 11 Installing: NumeriX®

11.1 Introduction

NumeriX valuation is available on the following platforms:

- Windows
- Linux 64b
- Solaris 64b.

You need to request a license key that is generated by NumeriX support which you install on the host computer.

Note: From WSS version 7.2.2.4 onwards, you should use NumeriX version 7.1.1 instead of version 6.1.2.

11.2 NumeriX license keys

11.2.1 License request Nx 6.1

Tell Wallstreet Systems the name of the end client, and the client's category. End client licenses fall into one of the following categories:

- Front office
- Back office
- Middle office.

11.2.1.1 Windows platforms

If you do not already have a license, please contact `license@numerix.com` and request a configuration file. When it arrives, save the configuration file to your Windows Desktop.

11.2.1.2 Unix platforms

For new installations only:

1. Evaluate the TRM environment
2. Run `nxprofile` (in the `$FK_HOME/bin` directory)
3. Email the resulting `profile.nxk` file as an attachment to `license@numerix.com`.

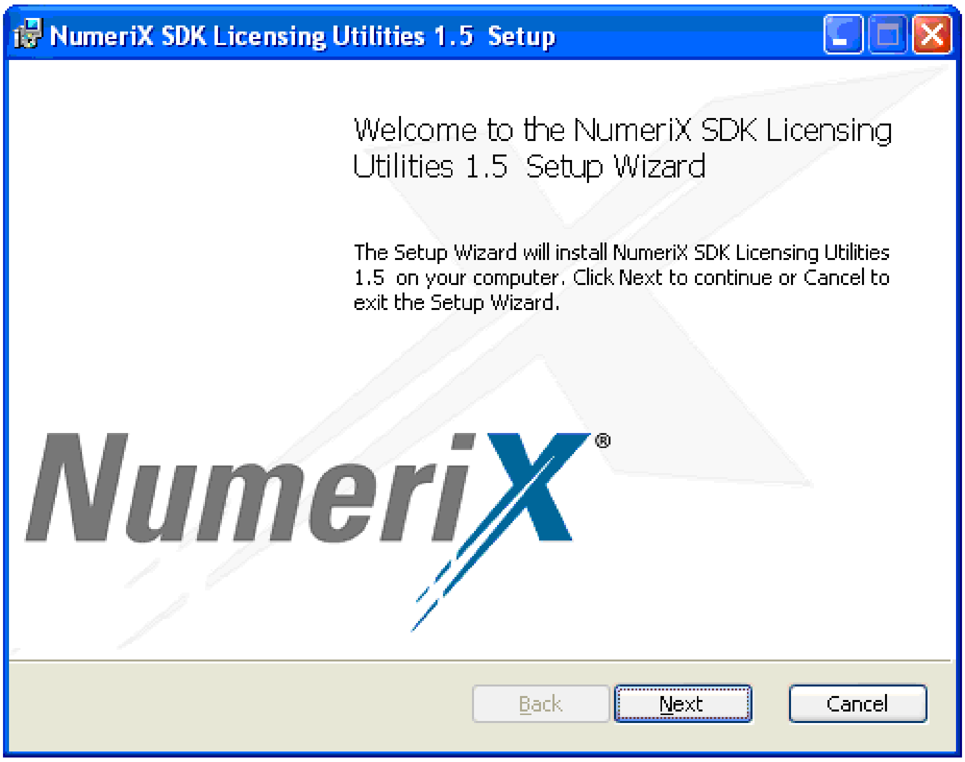
NumeriX will return your license key with installation instructions.

11.2.2 Installing the license and library

11.2.2.1 Installing the license under Windows

NumeriX Support will send a configuration file for NumeriX Kernel SDK. This configuration file is good for the installation on a single machine. If you are making additional installations, you should request additional configuration files from `license@numerix.com`.

- Save the file `*.config` to the Windows Desktop on the target machine(s).
- Run the installer `SDKLicenseUtilities_1_5.msi`, located in `%FK_HOME%\bin`.
- This installs the licensing Utilities. Administrator privileges are required to run this installer.
- Save the configuration file before running the NumeriX installer. The installer reads the configuration file and connects to the license server to obtain and install the license.

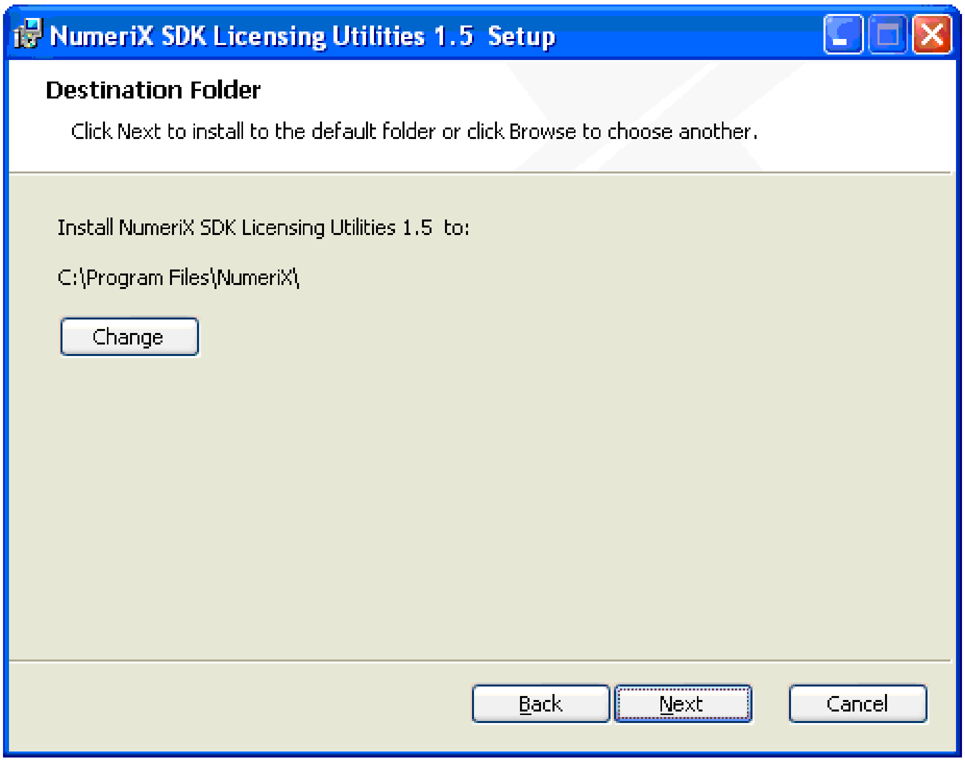


NumeriX SDK Licensing Utilities 1.5 Setup

Welcome to the NumeriX SDK Licensing Utilities 1.5 Setup Wizard

The Setup Wizard will install NumeriX SDK Licensing Utilities 1.5 on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

- Select the location where you want to install NumeriX Licensing Utilities:



NumeriX SDK Licensing Utilities 1.5 Setup

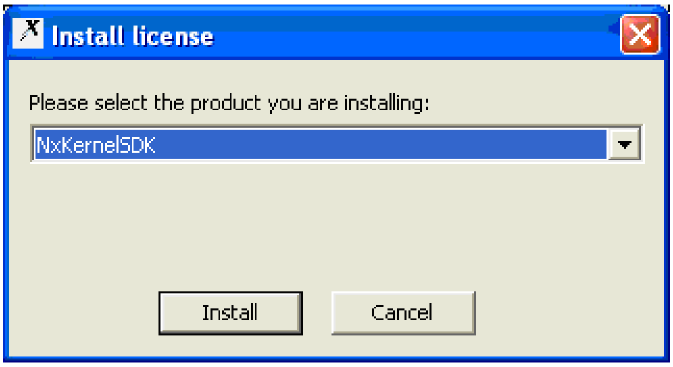
Destination Folder

Click Next to install to the default folder or click Browse to choose another.

Install NumeriX SDK Licensing Utilities 1.5 to:

C:\Program Files\NumeriX\

- Select the product NxKernelSDK:



Install license

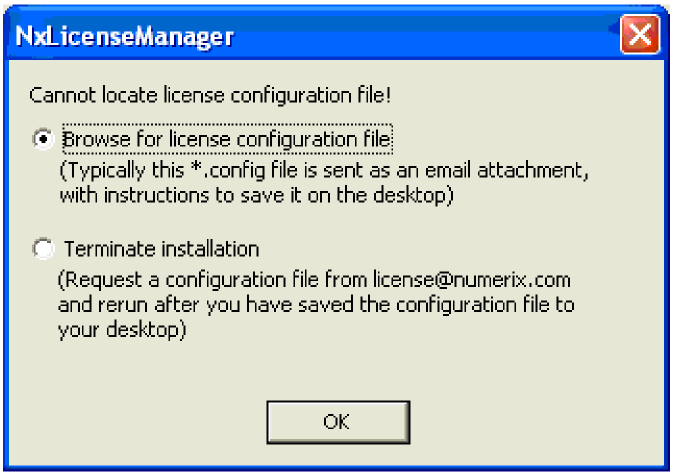
Please select the product you are installing:

NxKernelSDK

You must be able to connect, directly or indirectly, to the NumeriX global license server from the client machine(s) on which you intend to install NumeriX software.

If the machine connects directly to the Internet, the installer will find the *.config file and install the license silently. If the machine connects to the Internet via an http proxy server, you will see dialog boxes that guide you through the license installation via your http proxy server.

If you did not save the license file on your windows Desktop, you see this prompt:



NxLicenseManager

Cannot locate license configuration file!

● Browse for license configuration file
(Typically this *.config file is sent as an email attachment, with instructions to save it on the desktop)

○ Terminate installation
(Request a configuration file from license@numerix.com and rerun after you have saved the configuration file to your desktop)

If the procedure is successful, the license is activated, and NumeriX can be used.

11.2.2.2 Installing the license under Unix

NumeriX Support will send the `nxlib_license.nxk` file. The default is to look for the license file, `nxlib_license.nxk`, in the current directory. This can be overridden by setting either of the following environment variables:

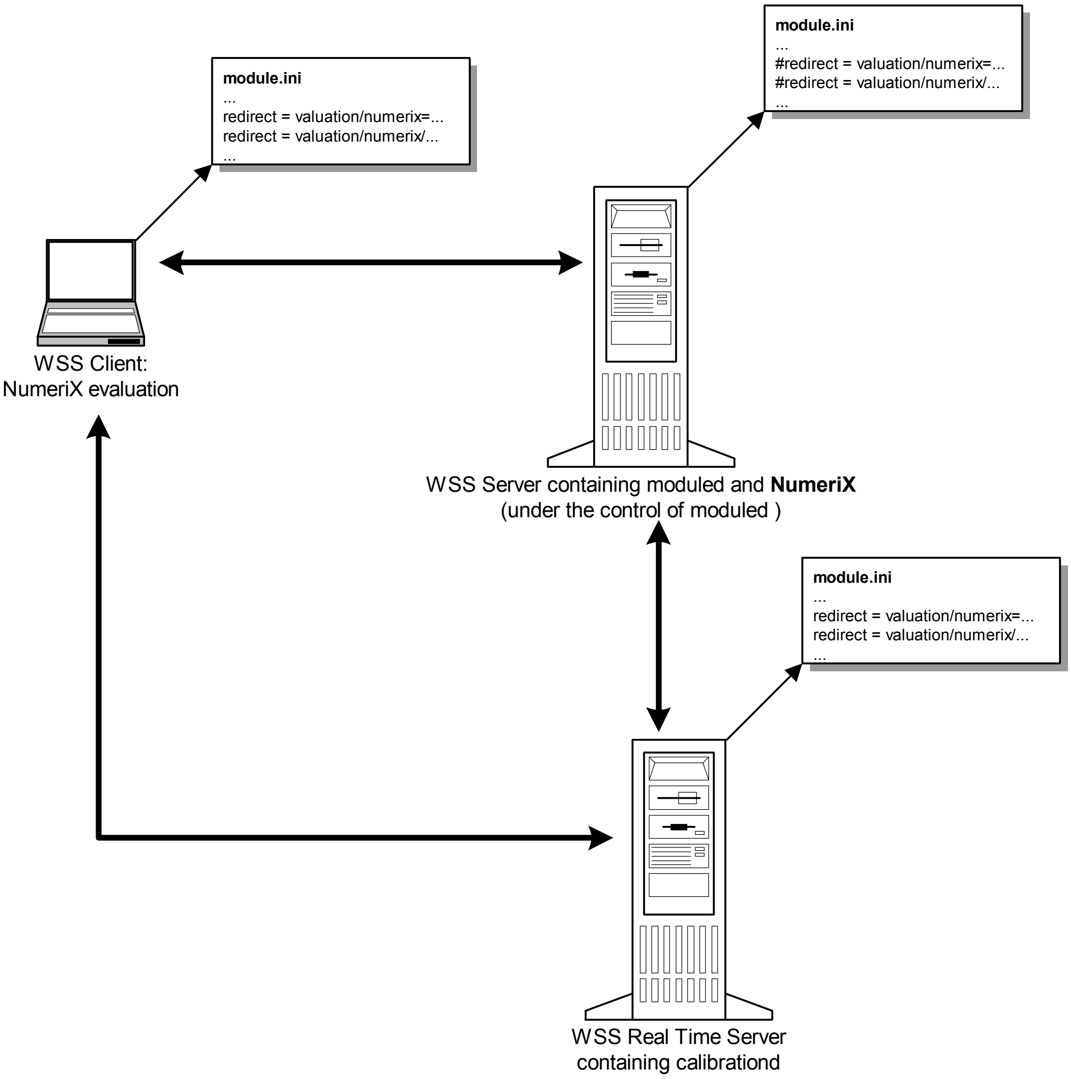
`NUMERIXLIBRARYLICENSEHOME` - looks for `nxlib_license.nxk` in this directory.

`NUMERIXLIBRARYLICENSEFILE` - specifies the full path and file name of license.

Suite Installer use the following folder if default parameters are used: `$FK_HOME/etc/numerix/license`.

11.3 Setting up a NumeriX server

If you have one or more computers in your WSS system that require NumeriX processing, you can configure the system so that calls are made to the server that hosts NumeriX via the WSS Real Time Server.


module.ini
...
redirect = valuation/numerix=...
redirect = valuation/numerix/...
...
module.ini
...
#redirect = valuation/numerix=...
#redirect = valuation/numerix/...
...
WSS Client:
NumeriX evaluation
WSS Server containing moduled and NumeriX
(under the control of moduled)
module.ini
...
redirect = valuation/numerix=...
redirect = valuation/numerix/...
...
WSS Real Time Server
containing calibrationd

To set this up, you should do the following:

1. Modify the `module.ini` file on some computers in the WSS system.
2. On the machine where Numerix is installed, enable the starting of the moduled process that controls NumeriX.
3. Edit the environment XML file used by Process Monitor so that the NumeriX moduled component can be started and stopped by Process Monitor.

11.3.1 Modify the module.ini files

On each computer participating on the WSS system, there is a file called `module.ini` located in `%FK_HOME%\etc`. On each computer *except the one where NumeriX is installed*, open the `module.ini` file and uncomment these two lines:

```
#redirect = valuation/numerix=loader/naming:valuation/numerix
#redirect = valuation/numerix/nx-adapter=loader/naming:valuation/numerix/nx-adapter
```

11.3.2 NumeriX and Process Monitor

To enable control of NumeriX via Process Monitor, you should edit a Process Monitor XML file. The services affected are `moduled-numerix` and `numerix-nx-adapter`. See the *WSS System Admin Guide* for a description of Process Monitor.

Open the relevant `etc\conf\<environment>.XML` file for Process Monitor, and add the text below:

```
<!--
    ModuleD - Numerix
  -->

  <process id="trm_moduled_numerix"
            desc="           TRM - Numerix daemon"
            agent="DevWinTRM" >
    <application pidfile="D:\WSS\wss_7212_dev\envs\dev\var\pid\moduled-numerix.pid" />
    <log file="D:\WSS\wss_7212_dev\envs\dev\var\logs/pmm/pmm_init-moduled-numerix.log"/>
    <manage control="AUTOMATIC">
      <start>
        <command path="D:\WSS\wss_7212_dev\envs\dev\bin/services/init.bat"
arguments="moduled-numerix py" />
        <variables>
          <set var="FK_LOGIN" value="[$(FK_MODULED_LOGIN)]" export="true"/>
          <set var="TS_LOG_FILE"
value="D:\WSS\wss_7212_dev\envs\dev\var\logs\trm/trm_moduled-numerix.log" />
          <set var="PMM_PID_FILE"
value="D:\WSS\wss_7212_dev\envs\dev\var\pid\moduled-numerix.pid" />
        </variables>
      </start>
      <stop kill="normal_then_brutal" />
      <dependencies>
        <requires process="trm_mdsd" />
      </dependencies>
    </manage>
  </process>

    <process id="trm_moduled_numerix-nx-adapter"
            desc="           TRM - Numerix NX daemon"
            agent="DevWinTRM" >
     <application
pidfile="D:\WSS\wss_7212_dev\envs\dev\var\pid\moduled-numerix-nx-adapter.pid" />
     <log
file="D:\WSS\wss_7212_dev\envs\dev\var\logs/pmm/pmm_init-moduled-numerix-nx-adapter.log"
/>
    <manage control="AUTOMATIC">
      <start>
        <command path="D:\WSS\wss_7212_dev\envs\dev\bin/services/init.bat"
arguments="moduled-numerix-nx-adapter py" />
        <variables>
          <set var="FK_LOGIN" value="[$(FK_MODULED_LOGIN)]" export="true"/>
          <set var="TS_LOG_FILE"
value="D:\WSS\wss_7212_dev\envs\dev\var\logs\trm/trm_moduled-numerix-nx-adapter.log" />
          <set var="PMM_PID_FILE"
value="D:\WSS\wss_7212_dev\envs\dev\var\pid\moduled-numerix-nx-adapter.pid" />
        </variables>
      </start>
      <stop kill="normal_then_brutal" />
      <dependencies>
        <requires process="trm_mdsd" />
      </dependencies>
    </manage>
  </process>
```

11.3.2.1 Enable the startup of moduled that controls NumeriX

Windows

If NumeriX is installed on a Windows server, create the following two files, and put them in `%FK_HOME%\bin`.

moduled-numerix.py

```
import os_utils
#=====================================================
# entry point
#=====================================================
if __name__ == "__main__":
os_utils.run_basic_app('moduled', '-m valuation/numerix -r')
```

moduled-numerix-nx-adapter.py

```
import os_utils
#=====================================================
# entry point
#=====================================================
if __name__ == "__main__":
os_utils.run_basic_app('moduled', '-m valuation/numerix/nx-adapter -r')
```

Unix

If NumeriX is installed on a Unix server, start moduled with the following commands:

```
$FK_HOME/sbin/moduled -m valuation/numerix -r
$FK_HOME/sbin/moduled -m valuation/numerix/nx-adapter -r
```

11.4 Testing Numerix

Follow these steps:

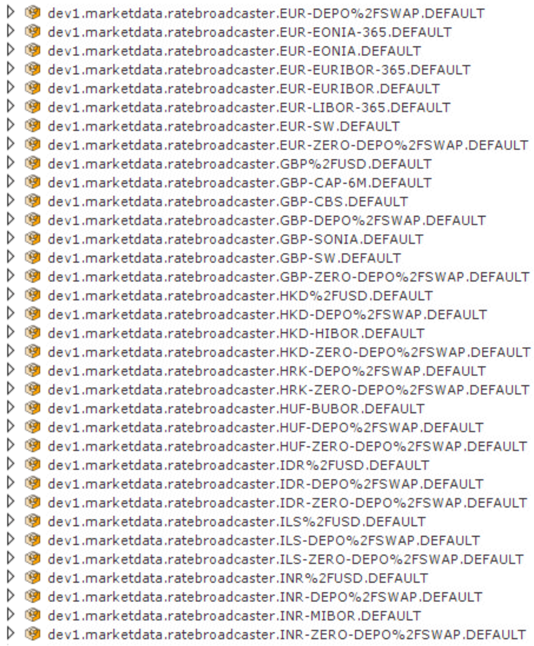
1. Retrieve a NumeriX-valued transaction in Transaction Manager.
2. Ensure that the model used for the valuation has been calibrated (in Calibration Monitor).
3. Run valuation for the transaction (**Options - Valuation Configuration**).

Software check: you can check the naming service with `nameclt` and `catior` to verify that the service exists, and is installed on the correct server.

Chapter 12 Installing: rate interfaces

12.1 Introduction

This section describes how to install and configure the Rate Interfaces components. These are are implemented as Onyx services, and are described in the table below:

Component name	Onyx service name	Description
Rate Feeder (Reuters)	`trmreuters`	Rate Feeder implements Reuters Foundation API for Java (RFA). When the service starts, it uses the Static Data Onyx service to load Market Info definitions entered via Market Info Source Editor. Rate Feeder is notified of any change to Market Info definitions. The list of Reuters Instrument Codes (RICs) to subscribe is refreshed after a certain period, besides being triggered by a static data message delivered through the message bus.
Rate Saver	`trmratesaver`	Rate Saver stores both the market data acquired from a provider (base data) and market data derived from them in a database. Rate Saver uses JDBC batching and multiple threads in order to achieve high throughput while minimizing the database server load.
MDSD Gateway	`trmmdsdgw`	MDSD Gateway connects Rate Interfaces components with mdsd. Currently it subscribes to Market Quotes coming from transd as a result of database operations (PendingPrices table) and to Static Data cache, invalidating messages which are used to purge the Static Data service caches.
Rate Broadcaster	`trmratebroadcaster`	Rate Broadcaster creates a set of topics. The topic name are derived from the instrument names as specified for the relevant Market Info definition. Owing to a limitation imposed on message bus topic names, the topic names must be URL-encoded. For instance, the topics used for EUR/USD rates will be named EUR%2FUSD.  dev1.marketdata.ratebroadcaster.EUR-DEPO%2FSWAP.DEFAULT dev1.marketdata.ratebroadcaster.EUR-EONIA-365.DEFAULT dev1.marketdata.ratebroadcaster.EUR-EONIA.DEFAULT dev1.marketdata.ratebroadcaster.EUR-EURIBOR-365.DEFAULT dev1.marketdata.ratebroadcaster.EUR-EURIBOR.DEFAULT dev1.marketdata.ratebroadcaster.EUR-LIBOR-365.DEFAULT dev1.marketdata.ratebroadcaster.EUR-SW.DEFAULT dev1.marketdata.ratebroadcaster.EUR-ZERO-DEPO%2FSWAP.DEFAULT dev1.marketdata.ratebroadcaster.GBP%2FUSD.DEFAULT dev1.marketdata.ratebroadcaster.GBP-CAP-6M.DEFAULT dev1.marketdata.ratebroadcaster.GBP-CBS.DEFAULT dev1.marketdata.ratebroadcaster.GBP-DEPO%2FSWAP.DEFAULT dev1.marketdata.ratebroadcaster.GBP-SONIA.DEFAULT dev1.marketdata.ratebroadcaster.GBP-SW.DEFAULT dev1.marketdata.ratebroadcaster.GBP-ZERO-DEPO%2FSWAP.DEFAULT dev1.marketdata.ratebroadcaster.HKD%2FUSD.DEFAULT dev1.marketdata.ratebroadcaster.HKD-DEPO%2FSWAP.DEFAULT dev1.marketdata.ratebroadcaster.HKD-HIBOR.DEFAULT dev1.marketdata.ratebroadcaster.HKD-ZERO-DEPO%2FSWAP.DEFAULT dev1.marketdata.ratebroadcaster.HRK-DEPO%2FSWAP.DEFAULT dev1.marketdata.ratebroadcaster.HRK-ZERO-DEPO%2FSWAP.DEFAULT dev1.marketdata.ratebroadcaster.HUF-BUBOR.DEFAULT dev1.marketdata.ratebroadcaster.HUF-DEPO%2FSWAP.DEFAULT dev1.marketdata.ratebroadcaster.HUF-ZERO-DEPO%2FSWAP.DEFAULT dev1.marketdata.ratebroadcaster.IDR%2FUSD.DEFAULT dev1.marketdata.ratebroadcaster.IDR-DEPO%2FSWAP.DEFAULT dev1.marketdata.ratebroadcaster.IDR-ZERO-DEPO%2FSWAP.DEFAULT dev1.marketdata.ratebroadcaster.ILS%2FUSD.DEFAULT dev1.marketdata.ratebroadcaster.ILS-DEPO%2FSWAP.DEFAULT dev1.marketdata.ratebroadcaster.ILS-ZERO-DEPO%2FSWAP.DEFAULT dev1.marketdata.ratebroadcaster.INR%2FUSD.DEFAULT dev1.marketdata.ratebroadcaster.INR-DEPO%2FSWAP.DEFAULT dev1.marketdata.ratebroadcaster.INR-MIBOR.DEFAULT dev1.marketdata.ratebroadcaster.INR-ZERO-DEPO%2FSWAP.DEFAULT

12.1.1 Operating system independent

As the services are implemented in Java, no specific configuration for Windows, Solaris, or Linux is needed. Installing Wallstreet Suite also installs Rate Interfaces services by default.

12.1.2 Integration with Process Monitor

The Onyx services are located under the **Onyx Rate Interfaces** node in the Process Monitor screen. The services are started as a part of the TRM module.

12.2 Configuring the rate interface services

The following property files are used to configure the rate interface services. The files `reuters.properties` and `rates.properties` are used exclusively by the Rate Interfaces services. The other property files are shared and used by other Onyx services.

12.2.1 reuters.properties (config.properties)

Make any changes to the file:
`<installation home>/sharedconf/components/reuters/config.properties`
and **not** to the `file`:
`<installation home>/envs/<env_name>/etc/onyx/properties/reuters.properties.`

When you perform a "Reconfigure" with the Wallstreet Suite Installer, a new `reuters.properties` file is written, based on the values in `config.properties`. Thus any changes made only to `reuters.properties` are overwritten.

property name in config.properties	Default	Description
`suite.installer.reuters.type`	`none`	Reuters service type. Valid values are: `none`, `omm` for the Open Model Messaging interface, and `md` for the MarketData interface. Use `none` to disable the service.
`suite.installer.reuters.serverList`	`localhost`	IP address or host name of the Reuters Market Data System (RMDS) server.
`reuters.portNumber`	`14002`	Reuters server port number. OMM: 14002 MarketData: 8101
`suite.installer.reuters.principalIdentity.name`	`guest`	RMDS user name.
`suite.installer.reuters.principalIdentity.appName`	`256`	Application identifier.
`suite.installer.reuters.prefs.userName`	`guest`	When using a Source Sink Library (SSL) connection, the SSL user’s name. (You may need to add this property to this file.)
`suite.installer.reuters.prefs.logFileName`	`none`	Additional RFA logging file. `none` is used to disable logging.
`suite.installer.reuters.defaultService`	`IDN_RDF`	The default service name used if none is provided in Market Info Source Editor.

12.2.2 rates.properties

Location: *<ONYX_CONFIG_DIR>*/properties/rates.properties

The file contains system properties which should not be modified.

12.2.3 orb.properties

Location: *<ONYX_CONFIG_DIR>*/properties/orb.properties

The file contains properties used to connect to the CORBA naming service. This is used by MDSD Gateway.

12.2.4 sybase.properties, oracle.properties, mssql.properties

Location: *<ONYX_CONFIG_DIR>*/properties/*<database_platform>*.properties

The file contains various properties used to establish a database connection. This is used by Rate Saver.

12.2.5 jms.properties

Location: *<ONYX_CONFIG_DIR>*/properties/jms.properties

The files contains properties used to configure connection to JMS broker (ActiveMQ server).

12.2.6 Reuters configuration files

The Reuters configuration files are:

- RSSL session/connection configuration file. for an example, see *12.7 Reuters configuration file* on page 123.
- Reuters data type definition file (enumtype.def)
- Reuters field definition file (RDMFieldDictionary)

The files enumtype.def and RDMFieldDictionary are provided by Reuters. They are stored in the *<ONYX_CONFIG_DIR>*/reuters/RDM directory of the Wallstreet Suite installation. These files should not require customizing. If you want to change the location where the files should be loaded from, use reuters.prefs.enumTypeFile and reuters.prefs.masterFidFile in the reuters.properties configuration file.

The RSSL session/connection configuration file is distributed as an internal file. Additional configuration options are added to the XML tree and the resulting document is then used for the session/connection configuration.

If you have a custom configuration file you wish to load, use the following properties in the reuters.properties configuration file:

- reuters.session.config.resourceClass
  the class used to load the resource. Must implement org.springframework.core.io.Resource interface. Examples:
  - org.springframework.core.io.FileSystemResource
    for file system based resources, for example /<path>/<file>
  - org.springframework.core.io.UrlResource
    for URL based resources, for example http://<hostname>/<file>
- reuters.session.config.resource
  the actual resource to load (a filename, a URL)
- reuters.session.namespace
  session namespace, for instance WSSNamespace::rsslSession (see *12.7 Reuters configuration file* on page 123)

- reuters.connection.namespace
  connection namespace, for instance WSSNamespace/Connections/rsslConnection (see *12.7 Reuters configuration file* on page 123)

These properties allow you to use a completely customized Reuters session/connection configuration.

12.3 Additional options

You can modify the Rate Interfaces behavior by modifying properties that can be supplied as additional properties in reuters.properties (for Rate Feeder component) or in rates.properties for all components. Another possibility is to specify the options directly as JVM options of the respective Onyx services (using -D<name>=<value> syntax).

Property name	Default	Description
dbWriter.threads	11	The number of concurrent threads used to access the database
filter.changePercent.default	0.01	The default value for Change Percent when not specified in Market Info Filter Editor
filter.limitThroughput.default	0	The default value for Limit Throughput when not specified in Market Info Filter Editor
filter.maximumPeriod.default	60	The default value for Maximum Period when not specified in Market Info Filter Editor
ratesaver.batchSize	300	The batch size for Rate Saver
reuters.subscriptionDelay	0	The time to delay between subscriptions to reduce CPU load at startup time
reuters.prefs.*		All properties beginning with reuters.prefs. are added under the connection node (WSSNamespace::rsslConnection) in the Reuters session/connection configuration file.

12.4 Prerequisites

The following components are necessary for Rate Interface in order to function properly.

- **Static Data service (Onyx)**

  The Static Data service is used to load Market Info definitions which are used to create subscriptions. When MDSD Gateway is running, Static Data service can be notified of any changes, and purges its static data caches.

  The Static Data service is also used to load additional details about static data objects which are used during the rate processing, including Scenarios.

  The Static Data service is used to fetch the Market Info Filter setup. Setup changes are also subject to MDSD Gateway notifications.

- **Security service (Onyx)**

  The Security service provides basic domain and object permission checking when accessing static data.

- **Message Delivery Service Daemon (mdsd)**

In a Rate Interfaces context, mdsd is used as a message dispatcher between static data editors and Onyx services. MDSD Gateway can connect to mdsd, and it transforms mdsd messages to messages that can be used within the Rate Interfaces messaging infrastructure.

- **MM Futures Chain service (a serviced instance)**

  A serviced instance must be running in order to subscribe to Futures Chain RICs. The service returns a list of RICs derived from an Instrument definition.

- **Transaction Daemon (transd)**

  In a Rate Interfaces context, transd is used as a bridge between a JMS broker (ActiveMQ) and non-message bus-based applications. Transd connects to a queue which contains rates to be passed to such applications, and passes them to MDSD.

- **Market Info Calculation Daemon (micd)**

  Market Info Calculation Daemon is now enhanced so that it consumes and produces rate messages directly from and to the message bus. It no longer uses MDSD for interprocess communication.

- **Naming service**

  The naming service is necessary for MDSD Gateway, which uses the naming service to connect to MDSD.

- **ActiveMQ server**

  This is the backbone of the system acting as a JMS broker (or message bus).

12.5 Hardware and Software Requirements

At peak load, Rate Saver may have several database transactions opened while processing hundreds of records in each transaction. To avoid problems with the log file segment (Sybase and MSSQL), make sure that the log file segment is big enough. The actual size depends on other database activity so it cannot be estimated. We recommend an additional 100MB of log file segment.

12.6 Troubleshooting

All Onyx services use log4j for logging. The log4j configuration file is located under *<ONYX_CONFIG_DIR>*/`log4j.xml`. The log file is specified in the `ONYX_LOG4J_LOG_FILE` environment variable. The log4j logging levels can be adjusted to acquire additional information from the running components.

The RMDS server sends many messages that report item status. Rate Feeder aggregates the messages and dumps the items' last status every ten minutes into a separate file. The file has the same name and location as the master log file except that it is suffixed with an additional extension: `.trmreuters`. If you have any doubts about a RIC validity, or if there is any problem receiving a specified item, this log file provides useful and simple information.

12.7 Reuters configuration file

This default configuration file is packaged in the `trmreuters` module. It can be partially altered by specifying `reuters.prefs.* properties` or can be replaced by a user-created file located elsewhere. This version contains only a subset of the original parameters that are relevant to Wallstreet Suite.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE preferences SYSTEM "http://java.sun.com/dtd/preferences.dtd">
<preferences EXTERNAL_XML_VERSION="1.0">
  <root type="user">
    <map />
    <node name="com">
      <map />
      <node name="reuters">
        <map />
        <node name="rfa">
          <map />
          <node name="_System">
            <map />
            <node name="EventQueue">
              <map>
                <entry key="batchingEnabled" value="true" />
                <entry key="batchSize" value="100" />
                <entry key="flushTimerInterval" value="50" />
              </map>
            </node>
          </node>
          <node name="WSSNamespace">
            <map />
            <node name="Connections">
              <map />
              <node name="rsslConnection">
                <map>
                  <entry key="connectionType" value="RSSL" />
                  <entry key="throttleEnabled" value="true" />
                  <entry key="throttleType" value="count" />
                  <entry key="throttleBatchCount" value="1" />
                  <entry key="throttleMaxCount" value="10" />
                  <entry key="tcp_nodelay" value="true" />
                </map>
              </node>
            </node>
            <node name="Sessions">
              <map />
              <node name="rsslSession">
                <map>
                  <entry key="connectionList" value="rsslConnection" />
                </map>
              </node>
            </node>
          </node>
        </node>
      </node>
    </node>
  </root>
</preferences>
```

12.7.1 Reuters: further information

for further information, log onto the Reuters support site at `https://customers.reuters.com/a/support.`

Appendix A Configuration Properties

The configuration properties located in `..\sharedconf` are used to configure all the XML and environment files required by Wallstreet Suite. Within the `sharedconf` folder are folders with variables specific to each module. The default properties for all modules have been carefully set to ensure a properly functioning integrated environment based on the Factory data. However, for special installations, these properties should be reviewed and changed.

Each environment can override the properties in `sharedconf`. Do this by placing a copy or partial copy of the relevant file in `sharedconf` into the environment's `conf` folder.

Note: For simple installations (for example, local machine, single user, demo environments) only the properties documented in the "key properties" sections need to be configured.

Regardless of platform, all properties for which paths are specified must use forward slashes `/`. The correct path syntax will be derived based on the installation platform.

A.1 Third Party Software Properties

Location: `..\sharedconf\3rdparty-software.properties`

A.1.1 Key properties

ts.urlpath.software.jdk.dir

The full path to the Java Developer Kit (JDK).

This JDK is installed as part of the Wallstreet Suite prerequisites (see *3.2 Installing the necessary third party software* on page 25.

Example: `C:\jdk.1.6.0_11 or /opt/jdk1.6.0_11`

ts.urlpath.software.db2.dir

(Required for a DB2 installation only.)

The full path to where DB2 is installed.

Example: `ts.urlpath.software.db2.dir=C:/dev/IBM/SQLLIB`

ts.urlpath.db2.data.dir

(Required for a DB2 installation only.)

The full path to where DB2 data is stored.

Example: `ts.urlpath.db2.data.dir=C:/`

ts.urlpath.software.oracle.dir

(Required for an Oracle installation only.)

The full path to where Oracle is installed.

This path is used to determine the location of the RDBMS binaries.

Example: `ts.urlpath.software.oracle.dir=c:/oracle/ora92` or `/opt/oracle`

ts.urlpath.software.sybase.dir

(Required for a Sybase installation only.)

The full path to where Sybase is installed.

This path is used to determine the location of the RDBMS binaries.

Example: `ts.urlpath.software.sybase.dir=c:/sybase` or `/opt/sybase`

ts.urlpath.software.mssql.dir

(Required for an MSSQL installation only.)

The full path to where MSSQL is installed.

This path is used to determine the location of the RDBMS binaries.

Example: `ts.urlpath.software.mssql.dir=C:/Program Files/Microsoft SQL Server/MSSQL$TS1`

ts.urlpath.mssql.data.dir

(Required for an MSSQL installation only.)

The full path to where MSSQL data is stored.

Example: `ts.urlpath.software.mssql.dir=C:/Program Files/Microsoft SQL Server/MSSQL$TS1`

ts.urlpath.software.weblogic.dir

(Required for a Weblogic installation only.)

The full path to where the BEA Weblogic server is installed.

Example: `ts.urlpath.software.weblogic.dir=C:/bea/wlserver_10.0` or `/opt/bea/wlserver_10.0`

Note that `wlserver_10.0` is a subdirectory of the BEA server installation.

ts.urlpath.software.websphere.dir

(Required for a WebSphere installation only. Available for a standalone CMM installation only.)

The full path to where the IBM WebSphere server is installed.

Example: `ts.urlpath.software.websphere.dir=C:/dev/IBM/WebSphere`

ts.urlpath.numeric.licence.dir

(Required for a NumeriX installation only.)

The full path to the NumeriX license.

Example : `C:\Numerix\license`

ts.urlpath.swift.alliance.dir

(Required for a SwiftAlliance and the Enterprise Swift Integration Adapter (Esiadapter) installation only)

The full path to the SwiftAlliance product.

Example: `C:\SWIFTAlliance`

A.2 Database Connection Properties

Location: `..\sharedconf\database.properties`

This file contains all the database connnection information suite. The modules use slightly different connection mechanisms depending mostly on the implementation technology (C++, Java, etc.). This database properties file contains all the connection parameters required to configure each of module specific configuration files.

The default file is configured to have all the modules located in one database/schema. In this configuration, only the first few properties need to be modified.

A.2.1 Key properties

A.2.1.1 Properties common to all three database types

db.name:

Database name for MSSQL and Sybase. For Oracle, this variable corresponds to the SID.

db.username:

The user to use for the default database connection and application servers.

Typically: `dbo` for Oracle, `fk` for MSSQL and Sybase.

db.password:

The password for the user specified in `db.username`.

All other usernames and passwords are initially set to their defaults. From version 7.2.1.2, these defaults are encrypted, see *Appendix H Password encryption* on page 159.

db.port:

Database server port.

- Oracle default: 1521
- MSSQL default instance: 1433
- MSSQL named instance: Needs to be determined by looking in the MSSQL Enterprise Manager

db.server:

DNS resolved host name of the database server, or the database server IP Address.

db.schemaowner:

Schema owner, this is case sensitive.

Typically: `DBO` for Oracle, and `dbo` for MSSQL and Sybase.

For user names and passwords see *A.3 User Properties on page 128*.

Note: If your CMM resides in a different database/schema, the CMM properties will have to be individually overridden.

db.type:

The Wallstreet Suite database type. Possible values are: `oracle`, `mssql`, `sybase`, `db2`.

db.version:

The database version.

- Oracle: `10g` or `11g` (default is `11g`)
- MSSQL: `2005` or `2008` (default is `2008`)
- Sybase: 12.5 or 15.0 (default is 12.5)

A.2.1.2 Oracle properties only

db.alias:

Database TNS alias, for example `sid.corp.trema.com`.

db.oracle.version

The version of Oracle installed. Possible values are: `10g` or 11g (default value is `11g`).

A.2.1.3 MSSQL properties only

db.instance:

The Microsoft SQL Server instance name.

db.mssql.version

The version of MSSQL installed. Possible values are: `2005` or `2008` (default value is `2008`).

A.3 User Properties

Location: `..\sharedconf\user.properties`

This file contains all "admin" level user names and passwords. From version 7.2.1.2, the passwords can be encrypted, see *Appendix H Password encryption* on page 159.

A.4 Host Properties

Location: `..\sharedconf\host.properties`

This file contains host information for the suite. All host names in this file should be resolvable by DNS. The default values are set to `localhost` which may not work in environments with multiple network interfaces. To ensure the hostnames are properly resolved use: `dig`, or `nslookup`.

A.4.1 Key properties

hostname

The DNS resolvable host name of the network interface to run services on. This hostname is used to create shortcut URLs, and define network connections.

hostip

The IP address corresponding to `hostname`. You can use the host name for this value, but to prevent DNS issues in the network installation we advise you to use the IP address instead.

A.4.2 Other properties

A.4.2.1 Secondary host properties

realtime.hostname

The DNS resolvable hostname for the realtime server. This only needs to be set if it is different from the hostname specified above. This is the host name for the server on which all the Wallstreet Suite real time processes run.

realtime.hostip

The IP address coresponding to `realtime.hostname`. You can use the host name for this value, but to prevent DNS issues in the network installation we advise you to use the IP address instead.

appserver.hostname

The DNS resolvable hostname for the web application server. Use this property if the J2EE application servers are running on a different host.

domain.name

The company domain name as accessed from client web browsers. The domain (or just the final part of the domain) name that must be resolvable by the DNS.

Example: `.corp.com`

The domain name is the unique name of a computer on the Intranet/Internet that distinguishes it from the other systems on the network. You can leave this blank, but this could cause security issues.

If a proxy server is set up in front of the Wallstreet Suite servers, set the domain name to the domain name of the proxy server.

A.4.2.2 Mail properties

smtp.mail.hostname

SMTP mail server hostname, used to send email notifications from the Wallstreet Suite.

smtp.mail.to

The default email address to send email notifications to. Use simple and not complex email addresses (eg `some.one@mailaddress.com` and not `Some One <some.one@mailaddress.com>`).

smtp.mail.from

The default from email address from which all notifications will be sent. This is normally an email address that can not be replied to. Use simple and not complex email addresses (eg `some.one@mailaddress.com` and not `Some One <some.one@mailaddress.com>`).

smtp.mail.subject

Default email subject. This will be used to form part of the the email subject for mail sent from the Wallstreet Suite.

smtp.mail.port

The mail port number.

A.4.2.3 Printer properties

printer.hostname

The network host to which the printer is attached.

printer.name

The name of the printer on the network host.

A.5 Application server properties

Location: `..\sharedconf\appservers.properties`

This file contains application server information for Wallstreet Suite.

A.5.1 Key properties

ts.appserver.type

The application server platform: either `tomcat` or `weblogic`.

Choosing `weblogic` requires `ts-weblogic` module activation.

Choosing `tomcat` requires `ts-tomcat` module activation.

ts.appserver.domain.name

The company domain name as accessed from client web browsers.

The domain (or just the final part of the domain) name that must be resolvable by the DNS.

Example: `.corp.com`

The domain name is the unique name of a computer on the Intranet/Internet that distinguishes it from the other systems on the network.

You can leave this blank, but this could cause security issues. However, if you use `localhost`, then you should leave this blank.

If a proxy server is set up in front of the Wallstreet Suite servers, set the domain name to the domain name of the proxy server.

ts.appserver.timezone

This value enables you to set a timezone, and must be a valid timezone. The default value is `GMT`.

A.5.2 Other properties

A.5.2.1 Secondary host properties

The following parameters enable the property configuration on each Wallstreet Suite application server. These properties include HTTP, HTTPS, URL parameters. They should be prefixed by the application server identifier:

- `ts.appserver.t1` is WebSuite
- `ts.appserver.t2` is TCMD
- `ts.appserver.t3` is ACM

Note: The WebSuite URL cannot be changed (`ts.appserver.t1.url.prefix`).

A.6 Weblogic properties

Location: `..\sharedconf\weblogic.properties`

A.6.1 Key properties

weblogic.admin.user

The Weblogic administrator’s user name. Default is `weblogic`.

weblogic.admin.passwd

The Weblogic administrator’s password. Default is `weblogic12`.

A.7 Process Monitor properties

Location: `..\sharedconf\components\pmm\config.properties`

A.7.1 Key properties

pmm.daemon.port

Port used by the Process Monitor Service Daemon (**pmsd**)

pmm.agent.port

Port used by the Process Monitor Agent Daemon (**pmad**).

A.8 Other properties

Other properties files can be found in the `..\sharedconf\` folder. All key properties in these files contain comments on their usage.

Appendix B Configuration data sheet

Use this table to provide configuration data for your Wallstreet Suite system. Either print it, or, if you have Acrobat Reader version 7, you can add information to this table using the Comment tools and save it.

Data required	Your values
Database type Either `Oracle`, `MSSQL`, or `Sybase` Example: `Oracle`	Value:
Database server machine name The computer name of the computer where the database server is installed. Example: `dbserver.corp.com`	Value:
Full path to where the database is installed Example: c:\oracle\oracle11	Value:
Database port number Typical values are: Oracle: `1521` MSSQL: `1433` Sybase: `2024`	Value:
Production database instance name MSSQL, Sybase: required if there are multiple instances of the database server.	Value:
Test database instance name MSSQL, Sybase: required if there are multiple instances of the database server.	Value:
Production database name Suitable user-friendly name for the production database.	Value:
Test database name Suitable user-friendly name for the test database.	Value:
Application server name The computer name of the computer where the Wallstreet Suite software is installed. Example: `ts_app.corp.com`	Value:
Company domain name The company domain name as accessed from client web browsers. Example: `.corp.com`	Value:

Data required	Your values
Application Server IP address The fixed IP address of the computer where the Wallstreet Suite software is installed. Example: `123.45.67.111`	Value:
Weblogic installation directory The Weblogic server installation full path. Example: `c:/bea/wlserver_10.0`	Value:
Package download directory Where you placed the Wallstreet Suite packages Example: `d:\wssuite_download\7.`*`n.n.n`*	Value:
Wallstreet Suite installation directory Example: `c:\wssuite\7`*`nnn`*	Value:
Database system password Oracle: password for the `sys` user MSSQL: password for the `sa` user	Value:
RDBMS client side software directory Example: `c:\oracle\ora92`	Value:
Java installation directory Example: `c:\Sun\SDK\jdk`	Value:
Python installation directory Example: `c:\python25`	Value:

Appendix C Enabling SDM on a non-SDM system

This chapter describes how to enable Static Data Management (SDM) on a system that was installed and configured without SDM. SDM is described in the *TRM System Administration Guide*.

If you are also installing the Cash Management Module (CMM), then you should read the CMM documentation, available from the Support site.

Warning: This section applies only to Wallstreet Suite version 7.1.4.1 or later. Do not attempt this procedure on a system before version 7.1.4.1.

C.1 Enabling SDM as part of an upgrade

If you are upgrading Wallstreet Suite, then you should complete all upgrade steps before enabling SDM. In other words, you should have an upgraded and fully-functional Wallstreet Suite before following the procedure below.

C.2 Enabling SDM

Follow these steps:

1. Back up your data.
2. Check the consistency of existing static data:

   SDM must not be enabled if there are inconsistencies in static data tables that will be SDM-enabled. You must check for inconsistent data by running this command (select the one that corresponds to your database):

   Oracle:

   ```
   cd ${FK_HOME}/share/oracle
   ./build -U … -P … -D … -k -E upgrade/sdm/1_check_consistency/\*.sql
   ```

   MSSQL:

   ```
   cd %FK_HOME%\share\mssql
   perl build.pl -U … -P … -D … -S … -k -E upgrade\sdm\1_check_consistency\*.sql
   ```

   Sybase:

   ```
   cd ${FK_HOME}/share/sybase
   ./build -U … -P … -D … -S … -k -E upgrade/sdm/1_check_consistency/\*.sql
   ```

   This prints to the screen any static data subentities that reference nonexistent master entities. For example, "Contacts" subentities in Client Editor; its master entity is "Client".

   If this command prints any inconsistencies, you can either:

- Fix them all manually in the database and keep rerunning the above command until its output is empty.

Or:

- Run this (Oracle only):

```
DECLARE
  r NUMBER;
  CR1 SYBEMU.REFCURSOR;
BEGIN
  r := CheckMasterConsistency(
         Parea =>'SDM',
         Pfor_report_p => 0,
         Premove_dangling_subentities_p => 1,
         CR1 => CR1);
END;
/
```

- Run this (Sybase only):

```
exec CheckMasterConsistency
           @area = 'SDM',
           @for_report_p = 0,
           @remove_dangling_subentities_p = 1
go
```

This tries to delete all unreferenced subentities that are reported by the command in this step (step *2.* on page 135). If this procedure call fails to delete some of them (for example because although they do not have a "master" entity and therefore should not exist, they can be referenced from other tables), you should manually correct the consistency of the data.

3. Create SDM-specific database objects

Run this command (select the one that corresponds to your database):

Oracle:

```
./build -U … -P … -D … -k -E upgrade/sdm/2_create_objects/\*.sql
```

MSSQL:

```
perl build.pl -U … -P … -D … -S … -k -E upgrade\sdm\2_create_objects\*.sql
```

Sybase:

```
./build -U … -P … -D … -S … -k -E upgrade/sdm/2_create_objects/\*.sql
```

4. Rebuild all procedures, triggers, views, indexes with SDM

Run this command, and note the `-w` option: this is very important. Select the command that corresponds to your database:

Oracle:

```
./build -U … -P … -D … -w -p all -i all -V all -k -v
```

MSSQL:

```
perl build.pl -U … -P … -D … -S … -w -p all -i all -V all -k -v
```

Sybase:

```
./build -U … -P … -D … -S … -w -p all -i all -V all -k -v
```

WSS installation without CMM: follow these steps

1. Copy data to SDM system tables, and move them to state FINAL.
2. Run this command (select the one that corresponds to your database):

   Oracle:

   ```
   ./build -U … -P … -D … -k -E
   upgrade/sdm/3_and_4_convert_data_and_make_final/\*.sql
   ```

   MSSQL:

   ```
   perl build.pl -U … -P … -D … -S … -k -E
   upgrade\sdm\3_and_4_convert_data_and_make_final\*.sql
   ```

   Sybase:

   ```
   ./build -U … -P … -D … -S … -k -E
   upgrade/sdm/3_and_4_convert_data_and_make_final/\*.sql
   ```

Warning: The build command above initializes SDM system tables so that the state of all entities is FINAL. It bypasses the SDM entity flow and works even when there are circular dependencies between entities. It cannot be run on installations with CMM because it does not "synchronize" SDM entities to CMM.

3. Edit your environment file for both server and client (located in `$FK_HOME/share/environments/`).
4. Add this line:

   ```
   set FK_MODULES = SDM
   ```

5. Using SDM State Editor, verify that your SDM entity state flow corresponds to an installation **without** CMM (all entities have to move from state VERIFY directly to state FINAL). See the *TRM System Admin Guide* for information about the SDM State Editor.

Wallstreet Suite installation with CMM: follow these steps

1. Run this command (select the one that corresponds to your database):

   Oracle:

   ```
   ./build -U … -P … -D … -k -E upgrade/sdm/3_convert_data/\*.sql
   ```

   MSSQL:

   ```
   perl build.pl -U … -P … -D … -S … -k -E upgrade\sdm\3_convert_data\*.sql
   ```

   Sybase:

   ```
   ./build -U … -P … -D … -S … -k -E upgrade/sdm/3_convert_data/\*.sql
   ```

2. Edit your environment file for both server and client (located in `$FK_HOME/share/environments/`).
3. Add this line:

   ```
   set FK_MODULES = CMM SDM
   ```

4. Using SDM State Editor, verify that your SDM entity state flow corresponds to an installation **with** CMM (entities that have to be copied to CMM move from state VERIFY to state SEND). See the *TRM System Admin Guide* for information about the SDM State Editor.

All static data entities that are SDM-enabled are now in state OPEN. Before you can work with them, you must move them to state FINAL (entities are during this process also "synchronized" or "copied" to CMM). You must manually "accept" all entities in the proper order (which depends on references between entities). You can do this either in Static Data Editors for each particular entity or in the SDM Admin application. See the chapter on Static Data in the *TRM System Admin Guide* for more details.

Wallstreet Suite installation with or without CMM

Under normal circumstances you should not use scripts that are located in `upgrade/sdm/4_make_final`. These scripts move all SDM managed entities from state OPEN to state FINAL. They bypass the SDM entity flow and work even when there are circular dependencies between entities.

These scripts can be used only where:

- You do not have CMM installed.
- All SDM managed entities in your database are in state OPEN (for example, you have loaded data to an SDM-enabled database using the DBLoader tool. See the chapter on transferring data with the DBLoader tool in the *WSS System Admin Guide*.

The `build` commands above call the procedure `BuildPortfolioTree` at the end. This procedure checks if the portfolio hierarchy is correctly set up. If it finds any inconsistencies, it prints them to the console and leaves the table `PortfolioTreeFlat` empty. Check that `BuildPortfolioTree` executed without any error messages and that the table `PortfolioTreeFlat` is not empty. If the table `PortfolioTreeFlat` is empty, the system cannot work correctly.

Appendix D The build script

D.1 Introduction

There are three scripts that perform operations on the Wallstreet Suite database:

- **`build.pl`**

  This script performs operation on TRM database objects

- **`acmbuild.pl`**

  This script performs operations on ACM database objects, and uses the same options as `build.pl`.

- **`cmmbuild.pl`**

  This script performs operations on CMM database objects, and uses almost the same options as `build.pl`. Differences are described in *D.3 cmmbuild.pl options* on page 142.

Important: This `cmmbuild.pl` script is new in 7.2 and is **not** the same as another script with the same name, described in *Appendix E Upgrading CMM from a version before version 7.1.3.2* on page 145 and *Appendix F The cmmbuild.pl tool (pre version 7.3.0)* on page 149.

D.2 build.pl options

The table below lists and describes the options that you can use with the WSS-provided `build.pl` script which can be found in one of the following directories:

- `$FK_HOME/share/oracle`
- `%FK_HOME%\share\mssql`
- `$FK_HOME/share/sybase`

Note: The table applies to all database types supported by the Wallstreet Suite. Please check the Oracle/Sybase/MSSQL columns to ensure that an option is valid for your database type.

Build script options	Oracle	Sybase	MSSQL	Description
`-U`	Yes	Yes	Yes	User name for connecting to the database.
`-P`	Yes	Yes	Yes	Password for connecting to the database.
`-S`	No	Yes	Yes	Name of the database server. Note: **MSSQL only** - if running this script on a remote desktop, add a \ immediately after the server name or its environment variable. For example: `-S %FK_DB_SERVER%\`

Build script options	Oracle	Sybase	MSSQL	Description
-D	Yes	Yes	Yes	Name of the database to which the script will connect. **Oracle only:** the entry from the file $ORACLE_HOME/network/admin/tnsnames.ora.
The remaining options are in alphabetical order.				
-a	Yes	Yes	Yes	Build everything (tables, views, rules, types, triggers and procedures).
-A	No	Yes	Yes	Modify existing tables. Use with the -t option.
-b	Yes	Yes	Yes	Prohibit logging in to the database. This option is useful if you want to create a file with all SQL commands (see the -o option).
-C	Yes	Yes	Yes	Build tables but keep existing data. Use with the -t option.
-d <default(s)>	No	Yes	Yes	Build system defaults (constant values that do not change).
-e	Yes	Yes	Yes	Print each executed statement and the time of execution.
-E <script(s)>	Yes	Yes	Yes	Execute given upgrade scripts. It is strongly recommended that you also use the options -v (print names of sql scripts which are executed) and -e (print each executed statement). Oracle version of build.pl turns on options -v and -e automatically. See the note at the end of this table. -E checks the validity of the upgrade path against the version stored in the database. When the path is not valid a warning is displayed, but the script execution is not affected. Do not use with other options apart from -U, -P, and -D. Upgrading an installation with SDM: do not specify the -w option when using the -E option.
-f	No	Yes	No	Dump transactions.
-g <object>	Yes	Yes	Yes	Build permissions for specified object (table, view, stored procedures or other database objects).
-G <table(s)>	No	Yes	No	Update identity gaps for given tables.
-i <index(es)>	Yes	Yes	Yes	Build indexes. Same syntax as -t.
-k	Yes	Yes	Yes	Do not stop on SQL errors. Normally, when the creation of any database object fails, the script reports the error and stops. When using this option, the script continues with the creation of all requested objects.
-l	Yes	Yes	Yes	Create default logins for dbo, ticket, batch, and limon.
-L <table(s)>	No	Yes	No	Update locking method for given tables.
-M	Yes	Yes	Yes	Turn on the options -a (build everything) and -b (prohibits logging in to the database).
-N <table(s)>	No	Yes	No	Update partitioning for given tables.

Build script options	Oracle	Sybase	MSSQL	Description
`-o <filename.sql>`	Yes	Yes	Yes	All SQL commands that create or change database objects are redirected to a text file instead of being sent to the database. This stops the script from creating or changing any database object, and enables you to review the SQL commands that would be sent. Note: You can use this option with option `-b` (do not connect to database).
`-p <procedure(s)>`	Yes	Yes	Yes	Build procedures and triggers. Same syntax as `-t`.
`-Q`	Yes	No	No	Compare the column order in the database with the column order in the table creation scripts. This option reports any differences and does not change anything in the database. Do not use any other options with `-Q` except the database connection information options and `-w`. You can use this option after an upgrade to verify that database tables have the same column order as specified in the table creation scripts.
`-s default/<site>`	Yes	Yes	Yes	`default`: the `-s` option runs all necessary scripts in the relevant directory below: • `$FK_HOME/share/oracle/setup` • `$FK_HOME/share/sybase/setup` • `%FK_HOME%\share\mssql\setup` <site>: same as above, except that after each script is run, its counterpart in is also run (if it exists) in the relevant directory below: • `$FK_HOME/site/<site>/share/oracle/setup` • `$FK_HOME/site/<site>/share/sybase/setup` • `%FK_HOME%\site\<site>\share\mssql\setup`
`-t <table(s)>`	Yes	Yes	Yes	Build tables by first truncating the table. Wildcards can be used. Use `all` to create all tables. It is not always possible to recreate a table because data integrity constraints may prohibit you from emptying the table. Use the script `build-data.pl` to either empty all table(s) first, or disable the integrity constraints.
`-T type(s)`	No	Yes	Yes	Build database types. Same syntax as -t above.
`-u`	Yes	Yes	Yes	Create default users and groups.
`-v`	Yes	Yes	Yes	Verbose. Script prints information about which objects are being built.
`-V <view(s)>`	Yes	Yes	Yes	Build views. Same syntax as `-t`.
`-w`	Yes	Yes	Yes	Enable the database for Static Data Manager support. This option must be used when installing a new TRM with SDM, and in all subsequent executions of `build.pl` after SDM has been installed. The exception is when upgrading an installation that has SDM: when using the `-E` option, there is no need to use `-w`. Note: This option has no effect when used with `cmmbuild.pl`.

Build script options	Oracle	Sybase	MSSQL	Description
`-x`	Yes	Yes	Yes	Prohibit automatically generated procedures from being created.
`-X`	Yes	Yes	Yes	Aligns database tables by adding, altering, renaming, and **dropping** columns so that a table in the database has the same structure as in the perl description. So if you have a column that was added by a plain SQL command (for example, a CSD), it will be lost (including its data). Therefore you must first modify the perl description so that it reflects your CSD.
`-y <filename>`	Yes	Yes	Yes	Alternative include-list file. Use the full path.
`-z`	Yes	Yes	Yes	Do not add static data to the tables. Use this option with the `-a`, `-r`, or `-t` options.

NOTES

MSSQL: By default, `build.pl` uses normal TRM login information. If the user does not have a login, or the user has no right to install forms, it is better to use the login parameters `-U` and `-P`.

Oracle: The -E option should only be used with upgrade scripts. Any other scripts should be run with `sqlplus`. When options `-v` and `-e` are turned on, you should redirect the output to a file: these options cause a lot of text to be generated.

D.3 cmmbuild.pl options

The `cmmbuild.pl` script supports most of the `build.pl` options described above, plus some extra ones. These differences are:

`-w` Disabled. If used, a message appears telling you that the option is disabled, and the script continues to process.

`-y` Not to be used. The script exits with a message telling you not to use this option.

`-nostorage` `cmmbuild` only: this option is switched on by default. If you specify this option anyway, you receive a confirmation message. (See `-withstorage` below for an explanation.)

`-withstorage` `cmmbuild` only: forces the use of a specific location where indexes or tables must be created. (This option switches off the `-nostorage` option.) See `storage.pl` in the `modules/cmm` folder for details.

D.4 Examples

Note: All examples below assume that you do not have SDM installed. If you do, then all `build` statements must include the `-w` option.

Change the directory containing the build script (pick one of the following):

```
cd ${FK_HOME}/share/oracle
```

```
cd ${FK_HOME}/share/sybase
```

```
cd %FK_HOME%\share\mssql
```

Build a complete database and log the build messages to a file:

Oracle

```
./build -U dbo -P <password> -D $FK_DATABASE -a -v -k > $DATABASE_BUILD_LOG 2>&1
```

MSSQL

```
perl build -U dbo -P <password> -S %FK_DB_SERVER% -D %FK_DATABASE% -a -v -k >
%DATABASE_BUILD_LOG% 2>&1
```

Sybase:

```
./build -U dbo -P <password> -S $FK_DB_SERVER -D $FK_DATABASE -a -v -k >
$DATABASE_BUILD_LOG 2>&1
```

Build all indexes:

Oracle

```
./build -U dbo -P <password> -D $FK_DATABASE -i all -v
```

MSSQL

```
perl build -U dbo -P <password> -S %FK_DB_SERVER% -D %FK_DATABASE% -i all -v
```

Sybase:

```
./build -U dbo -P <password> -S $FK_DB_SERVER -D $FK_DATABASE -i all -v
```

Build the procedure "ProcessSellings":

Oracle

```
./build -U dbo -P <password> -D $FK_DATABASE -p ProcessSellings -v
```

MSSQL

```
perl build -U dbo -P <password> -S %FK_DB_SERVER% -D %FK_DATABASE% -p ProcessSellings
-v
```

Sybase:

```
./build -U dbo -P <password> -S $FK_DB_SERVER -D $FK_DATABASE -p ProcessSellings -v
```

Build the procedure "AccountingEvent" table, but keep old date:

Oracle

```
./build -U dbo -P <password> -D $FK_DATABASE -C -t AccountingEvent -v
```

MSSQL

```
perl build -U dbo -P <password> -S %FK_DB_SERVER% -D %FK_DATABASE% -C -t
AccountingEvent -v
```

Sybase:

```
./build -U dbo -P <password> -S $FK_DB_SERVER -D $FK_DATABASE –C -t AccountingEvent –
v
```

Create the sql code for a procedure:

Oracle

```
./build -U dbo -P <password> -D $FK_DATABASE –p ProcessSellings –v –o
ProcessSellings.sql
```

MSSQL

```
perl build -U dbo -P <password> -S %FK_DB_SERVER% -D %FK_DATABASE% –p ProcessSellings
–v –o ProcessSellings.sql
```

Sybase:

```
./build -U dbo -P <password> -S $FK_DB_SERVER -D $FK_DATABASE –p ProcessSellings –v –
o ProcessSellings.sql
```

Appendix E Upgrading CMM from a version before version 7.1.3.2

Before CMM 7.1.3.2, organizations running CMM on an Oracle database used the CMM database tool to maintain and upgrade the CMM database. As of CMM 7.1.3.2, these customers must upgrade to a new tool: the `cmmbuild.pl` database build script. This tool provides for better usability and improves integration of CMM with the other modules of Wallstreet Suite.

The remainder of this appendix explains how to upgrade to `cmmbuild.pl`.

E.1 Applying required database update

Before completing the steps in this appendix, complete a full backup of the Wallstreet Suite database. It is always good practice to complete backups of the database in case users make error requiring database recovery.

The following scripts are required to prepare the database for use with `cmmbuild.pl`:

- `upgrade_from_7131_tablespaces_as_sys.sql`
- `upgrade_from_7131_procedure_as_owner.sql`
- `cmmdbo_rights.sql`.

These files are delivered with CMM 7.1.3.2.

Note: As of version 7.1.3, the CMM database is case sensitive. Be aware of this when upgrading and completing other tasks with the database.

E.1.1 Adding the required tablespaces

In some situations, the required tablespaces may already exist and, therefore, no further action is required. (Review the `upgrade_from_7131_tablespaces_as_sys.sql` script with your organization’s DBA to confirm whether the tablespaces do or do not exist.)

If the tablespaces do not exist:

1. Update the path examples in the `upgrade_from_7131_tablespaces_as_sys.sql` script to work with your organization’s installation.
2. Log into the existing CMM database as the sys user.
3. Execute the script.
4. Confirm that the script executed successfully and that the tablespaces were created.

   If the script failed, determine and resolve the cause and then rerun the script.

E.1.2 Updating the permissions of the schema owner

The sample script `cmmdbo_rights.sql` is valid for installations where the cmmdbo database user is the schema owner for the CMM database objects. In some installations, the schema owner may be the cmm_objects database user or another database user. If this is the case in your installation, change each occurrence of "cmmdbo" in the `cmmdbo_rights.sql` script with the name of the schema owner before completing the following procedure.

To update the permissions of the schema owner:

1. Log into the existing CMM database as the sys user.
2. Execute the `cmmdbo_rights.sql` script.
3. Confirm that the script executed successfully.

E.1.3 Adding the required procedure and patch data

To add the required procedure and patch data:

1. Log into the existing CMM database as the schema owner.
2. Execute the `upgrade_from_7131_procedure_as_owner.sql` script.
3. Ensure that the script executed successfully.

E.2 Installing and executing cmmbuild.pl

Before completing the procedure in this section, you must upgrade the TRM application to 7.1.3.2. (`cmmbuild.pl` is installed onto the existing TRM application and depends on version 7.1.3.2.)

E.2.0.1 Required files

The following file is required to install `cmmbuild.pl` in an integrated environment:

- `cmm-buildpl_7.1.3.2-r00XX_noarch.zip`.

Note: This file is delivered with CMM 7.1.3.2.

E.2.0.2 Procedure

To install `cmmbuild.pl` in an integrated environment:

1. Extract the `cmm-buildpl_7.1.X.X-r0xxx_noarch.zip` file into the `$FK_HOME` folder on top of the existing `share/oracle` folder structure.
2. Evaluate the environment.

   The environment is configured through environment variables defined in configuration files located in `$FK_HOME/share/environments`.

   Run your system's `eval` command to set the environment variables. The following is an example (in Solaris):

   ```
   eval `/usr/trema/fk71/bin/environ -e fkprod_ora -h /usr/trema/fk71 -f`
   ```

3. Do the following:

   – If you are installing on a Windows server, set the following environment variables:

   ```
   set PERL_HOME=C:\apps\cmm_buildpl\perl5.8.6
   set PATH=%PERL_HOME%\bin;%PATH%
   set PERL5LIB=%PERL_HOME%\lib
   ```

   (The paths are examples only.)

   – If you are installing on a Unix server, ensure the `LD_LIBRARY_PATH` includes `$PERL_HOME/lib`.

4. Check the connection to the Oracle database:

Operating system	Command
Windows	`sqlplus dbo/<password>@%FK_DB_SERVER%`
Unix	`sqlplus <schema_owner>/<password>@$FK_DB_SERVER`

(Replace `<password>` with the schema owner’s password.)

5. If you connected successfully in step 4, type "exit" and press the Enter key.
6. Navigate to the folder containing `cmmbuild.pl` (`$FK_HOME/share/oracle`).
7. Apply the latest database patches by executing `cmmbuild.pl`:

```
perl cmmbuild.pl -U <schema_owner> -P <password> -D $FK_DB_SERVER -p -v
```

Appendix F The cmmbuild.pl tool (pre version 7.3.0)

F.1 cmmbuild.pl options

Important: This refers to the `cmmbuild.pl` tool that should be used to upgrade the CMM part of the Wallstreet Suite database from a version prior to 7.3.0 up to version 7.3.0. From version 7.3.0 onwards, use the tool described in *Appendix D The build script* on page 139.

The table below lists and describes the options that you can use with the WSS-provided `cmmbuild.pl` script which can be found in the `$FK_HOME/share/oracle` directory.

Note: The table applies to all database types supported by the Wallstreet Suite. Please check the Oracle/Sybase/MSSQL columns to ensure that an option is valid for your database type.

Build script options	Oracle	Sybase	MSSQL	Description
`-U` (`-username`)	Yes	No	No	User name for connecting to the database.
`-P` (`-password`)	Yes	No	No	Password for connecting to the database.
`-S`	No	No	No	Name of the database server. Note: **MSSQL only** - if running this script on a remote desktop, add a `\` immediately after the server name or its environment variable. For example: `-S %FK_DB_SERVER%\`
`-D` (`-database`)	Yes	No	No	Alias of the database to which the script will connect. Oracle: the entry from the file `$ORACLE_HOME/network/admin/tnsnames.ora`.
`-v (-verbose)`	Yes	No	No	Verbose. Script prints information about which objects are being built.
`-o` <filename.sql> (`-output-file`) <filename.sql>	Yes	No	No	All SQL commands that create or change database objects are redirected to a text file instead of being sent to the database. This stops the script from creating or changing any database object, and enables you to review the SQL commands that would be sent. Note: You can use this option with option `–b` (do not connect to database).
`-a` (`-all-objects`)	Yes	No	No	Build everything (tables, views, rules, types, triggers and procedures).
`-m` (`-migrate-data`)	Yes	No	No	Perform a data migration from a 6.5.X version of CMM. This option requires –link-user, -link-passwd and -link-name
`-link-user`	Yes	No	No	User name for connecting to the source database during a 6.5.X migration. Usually set to auros_objects.

Build script options	Oracle	Sybase	MSSQL	Description
`-link-passwd`	Yes	No	No	Password for link-user for connecting to the source database during a 6.5.X migration.
`-link-name`	Yes	No	No	Alias of the database used in the migration. This is the 6.5.X database. Please note, the tnsnames entry must exist on the target database server, not the client. Oracle: the entry from the file `$ORACLE_HOME/network/admin/tnsnames.ora`.
`-p` (`-patch-database`)	Yes	No	No	Apply the database patches to the database. This option is not required when the –a (-all-objects) option is used.
`-r` (`-remove-objects`)	Yes	No	No	Remove the CMM specific database objects. This is not currently supported.
`-c` (`cmm-specific`)	Yes	No	No	This parameter must be used in all cases for CMM standalone installations
`-k`	Yes	Yes	Yes	Do not stop on SQL errors. Normally, when the creation of any database object fails, the script reports the error and stops. When using this option, the script continues with the creation of all requested objects.
`-y` <filename> (`-include-list`) (`-i`)	Yes	Yes	Yes	Alternative include-list file. Use the full path.

F.1.1 Examples

Build a complete database and log the build messages to a file. Suppress errors:

```
perl cmmbuild.pl -U dbo -P <password> -D $FK_DATABASE -a -v -k > $DATABASE_BUILD_LOG
2>&1
```

Build a complete database and log the build messages to a file. Load the CMM stand-alone data:

```
perl cmmbuild.pl -U dbo -P <password> -D $FK_DATABASE -a -v -cmm-specific >
$DATABASE_BUILD_LOG 2>&1
```

Build a complete database and log the build messages to a file. Migrate to a new integrated Wallstreet Suite installation:

```
perl cmmbuild.pl -U dbo -P <password> -D $FK_DATABASE -a -v –m –link-user
auros_objects –link-passwd alterna –link-name db69_julie > $DATABASE_BUILD_LOG 2>&1
```

Build a complete database and log the build messages to a file. Migrate an existing CMM database to a new CMM stand-alone installation:

```
perl cmmbuild.pl -U dbo -P <password> -D $FK_DATABASE -a -v -c -m -link-user
auros_objects -link-passwd alterna -link-name db69_julie > $DATABASE_BUILD_LOG 2>&1
```

Note: The `link-user` field is set to the schema owner in the CMM 6.5.X database; the value of the `link-passwd` field, the corresponding user's password; and the value of `link-name` field, the Oracle `TNSnames` connection string for the CMM 6.5.x database on the target database server. These values **must** be tested before proceeding with the data migration using a utility such as `sqlplus`.

F.1.2 Post migration: tables with less rows

When interpreting the results of a migration, you may notice that several tables have additional rows. This is common; however, you need to analyze the sections of the logs that indicate tables with less rows more carefully. The expected results should be similar to the following:

Table name	CMM 6.5.x	CMM 7.1.x	Rows missing
PATCHLOG	125	2	123
TABLESEQUENCES	29	0	29
USEROPTIONS	2797	2790	7
DBVERSION	1	0	1

If there are any other tables with less rows in version 7.1.x than in version 6.5.x or if you have any other questions or concerns regarding the log files, contact Wall Street Systems.

Appendix G Transferring data from Sybase to Oracle

G.1 Introduction

This section describes how to export Wallstreet Suite data from a Sybase database, then import it into an Oracle database, using the tools provided in the Wallstreet Suite installation package.

G.1.1 Using a Wallstreet Suite 6.5 database

If this is part of an upgrade from Wallstreet Suite 6.5 to 7.1:

- You must have a working version 7.1 environment.
- First perform the Sybase-Oracle data transfer on the version 6.5 database as described in this section.
- Then perform the Oracle database upgrade to version 7.1.

G.1.2 Summary of steps

The procedure consists of:

1. Creating an empty Oracle database
2. Dropping the indexes in the Oracle database
3. Exporting the Sybase data to flat files
4. Creating database tables (6.5 only)
5. Converting data
6. Importing data into Oracle
7. Rebuilding indexes
8. Final tasks and verification.

G.2 Creating an empty Oracle database

Create an empty Oracle database by following the steps in *4.2.3 Oracle database installation* on page 28.

G.3 Dropping all indexes in the target database

Loading data into the database is much faster without indexes, and you can rebuild them after the data is loaded.

Here is an example, using a shell:

```
sqlplus ${ORACLE_USER}/${ORACLE_PASSWORD}@${ORACLE_DATABASE} << EOS
set echo on
set serveroutput on size 1000000
begin
  for I in (
    select * from USER_INDEXES
    -- LOB indexes can not be dropped (ORA-22864). Also they are not
    -- indexes created by user, but by Oracle internally.
    where index_type != 'LOB'
  ) loop
    BEGIN
      DBMS_OUTPUT.put_line('Dropping index: ' || I.index_name);
      EXECUTE IMMEDIATE 'drop index ' || I.index_name;
    EXCEPTION when others then
      DBMS_OUTPUT.put_line('Error: ' || SQLERRM);
    END;
  end loop;
end;
/
exit
EOS
```

G.4 Exporting data from Sybase

1. Create directory where exported data will be stored (for example `/tmp/_sybase_data`):

   ```
   cd /tmp
   
   mkdir _sybase_data
   
   cd _sybase_data
   ```

2. Run the `eval` command for the Sybase environment.
3. Run the script to copy data out from Sybase into BCP files:

   Version 6.5:

   a. Log on as user `fkadmin`.
   b. Run this script:

      ```
      perl ${FK_HOME}/share/common/fkbcp.pl -S ${SYBASE_SERVER} -D
      ${SYBASE_DATABASE} -U ${SYBASE_USER} -P ${SYBASE_PASSWORD} -w out -T sybase
      -c"|||" -r"#|#" -n 1
      ```

Note: Replace ***FK_HOME*** above with the 7.1 `FK_HOME` path, because although you are using the version 6.5 environment, you should be using the 7.1 version of `fkbcp.pl`.

   Version 7.1

   ```
   perl ${FK_HOME}/share/common/fkbcp.pl -S ${SYBASE_SERVER} -D ${SYBASE_DATABASE}
   -U ${SYBASE_USER} -P ${SYBASE_PASSWORD} -w out -T sybase -c"|||" -r"#|#" -n 1 -J
   utf8
   ```

The log of the export of data is now in the file `bcpout.res` in the current directory.

G.4.1 Copying large tables from Sybase

Owing to file size restrictions in the BCP utility in Sybase, some tables could exceed the maximum file size of 2GB that can be handled by the utility. These tables must be handled separately:

1. Identify the tables whose data requires the file to be larger than 2GB. The command below lists all files in order of size with the largest files at the end of the list.

   ```
   ls -lsh | sort -n
   ```

2. For each table that has reached the 2GB limit, use the Sybase BCP command to copy the tables out in stages and concatenate the data into a single file. Use the same procedure as shown in this example for the Cashflow table:

   Cashflow

   ```
   $SYBASE/$SYBASE_OCS/bin/bcp sybase database..Cashflow out Cashflow.1.dat -U fk -P
   fk password -S sybase server -L 3000000 -t"|||" -r"#|#" -c -J utf8

   $SYBASE/$SYBASE_OCS /bin/bcp sybase database..Cashflow out Cashflow.2.dat -U fk
   -P fk password -S sybase server -F 3000001 -L 6000000 -t"|||" -r"#|#" -c -J utf8

   $SYBASE/$SYBASE_OCS /bin/bcp sybase database..Cashflow out Cashflow.3.dat -U fk
   -P fk password -S sybase server -F 6000001 -t"|||" -r"#|#" -c -J utf8

   cat Cashflow.1.dat > Cashflow.dat

   cat Cashflow.2.dat >> Cashflow.dat

   cat Cashflow.3.dat >> Cashflow.dat
   ```

G.5 Creating database tables (version 6.5 only)

Note: This section applies to a Wallstreet Suite version 6.5 conversion only.

Create new, empty version 6.5 database tables in Oracle. This will be the target database in the data transfer. This can be done using the `build` command in Wallstreet Suite 6.5. However, by default, the `build` command creates some initial static data that is not needed, so you must use the `-Z` option in the `build` command.

1. Log in as user `fkadmin`.
2. Run the `eval` command for the Wallstreet Suite 6.5 Oracle environment:

   ```
   eval `/usr/trema/fk65/bin/environ -e environment name`
   ```

3. Run the `build` command to create new and empty database tables:

   ```
   cd /usr/trema/fk65/share/oracle

   build -U dbo -P password -D database -az
   ```

4. If GLM was used, create empty GLM tables in the Wallstreet Suite version 6.5 Oracle database:
   a. Log in as user `fkadmin`.
   b. Run the `eval` command for the Wallstreet Suite version 6.5 Oracle environment:

      ```
      eval `/usr/trema/fk65/bin/environ -e environment name`
      ```

   c. Run the `glmbuild` command to create new and empty database tables:

      ```
      cd /usr/trema/glm6.5.2.2b/share/oracle
      ```

```
perl ./glmbuild.pl -U dbo -P trema -D database name -a  -v -k >
glmbuildnewdb.log 2>&1 &
```

G.6 Converting data

Convert data from Sybase format (output from `bcp`) to Oracle format, for input to `sqlloader`:

1. Create directory where converted data will be stored (for example `/tmp/_oracle_data`):

   ```
   cd /tmp
   ```

   ```
   mkdir _oracle_data
   ```

2. Run the `eval` command for the version 7.1 Oracle environment.
3. Run the `sqlloader` utility:

   ```
   cd ${FK_HOME}/share/oracle/migration
   ```

   ```
   perl bcp2sqlldr.pl -v -D ${ORACLE_DATABASE} -U ${ORACLE_USER} -P
   ${ORACLE_PASSWORD} -d "/tmp_sybase_data" -o "/tmp/_oracle_data" -B
   "${FK_HOME}/share/sybase"
   ```

   The last step also produces a command file `sqlldrcmd.sh` in the directory `/usr/trema/fk71/share/oracle/migration`. This command file will be used in the next step, where the files are loaded into the oracle database.

G.7 Importing data into Oracle

1. Tell `sqlldr` that data is encoded in UTF8.

   ```
   export NLS_LANG=_.UTF8
   ```

2. Run the script that was produced by the tool that converted the data. This script runs `sqlldr` for all data files:

   ```
   cd ${FK_HOME}/share/oracle/migration
   ```

   When the tool is run on Unix, it generates the file `sqlldrcmd.sh`:

   ```
   ./sqlldrcmd.sh
   ```

Note: When the tool is run on Windows, it generates the file `sqlldrcmd.bat`.

3. Look in the current directory for the subdirectory called `bad`. This contains logs of any unsuccesfully loaded tables.

G.8 Rebuilding indexes

Remember that we previously dropped all indexes, so rebuild them now:

```
cd ${FK_HOME}/share/oracle/
```

```
perl build.pl -i all -D ${ORACLE_DATABASE} -U ${ORACLE_USER} -P ${ORACLE_PASSWORD} -k
-v -w
```

Warning: Option **-w** should be used **only** if the database is SDM-enabled.

G.9 Rebuilding GLM indexes (version 6.5 only)

Note: This section applies to a Wallstreet Suite version 6.5 conversion only.

If your Wallstreet Suite 6.5 includes GLM, then you must rebuild all indexes for the GLM module. This is done using the `glmbuild.pl` command that comes with the GLM package.

1. Run the `glmbuild.pl` command:

   ```
   cd /usr/trema/glmn.n.n.n/share/oracle

   perl glmbuild.pl -U dbo -P password -D database -i all -v -k
   ```

2. Ignore the error message in the file.

G.10 Final tasks after data import

1. Execute an SQL script that computes statistics on all tables, recreates sequences that emulate identity columns, recreates timestamps, compiles invalid objects, and re-enables disabled triggers:

   ```
   ${FK_HOME}/share/oracle/migration/after-load-db.sql
   ```

2. Compute the database statistics:

   ```
   sqlplus ${ORACLE_USER}/${ORACLE_PASSWORD}@${ORACLE_DATABASE} << EOS

   execute dbms_utility.analyze_schema('DBO','COMPUTE');

   exit

   EOS
   ```

3. Verify the transfer:

   a. Check for errors in the log files:

      ```
      cd $FK_HOME/share/oracle/migration/log

      grep ORA *
      ```

   b. Run a count on each table in Sybase and a count on each table in Oracle. Make sure the number of rows in each table is correct.

Appendix H Password encryption

H.1 Encrypting passwords

Note: Password encryption was implemented from version 7.2.1.2 onwards.

To provide security, the password store in Suite Installer is encrypted. This means that the passwords defined in following places are encrypted:

- Suite Installer configuration:
  `$WSS_HOME/shared-conf/user.properties`
- Database User configuration:
  `$FK_HOME/share/<database_type>/config.pl`
- Process Monitor configuration:
  `<pmm_installation_folder>/etc/conf/private/env_<environment_name>.secret`

You should create encrypted passwords in your `site` directory using the encrypt application delivered with Suite Installer:

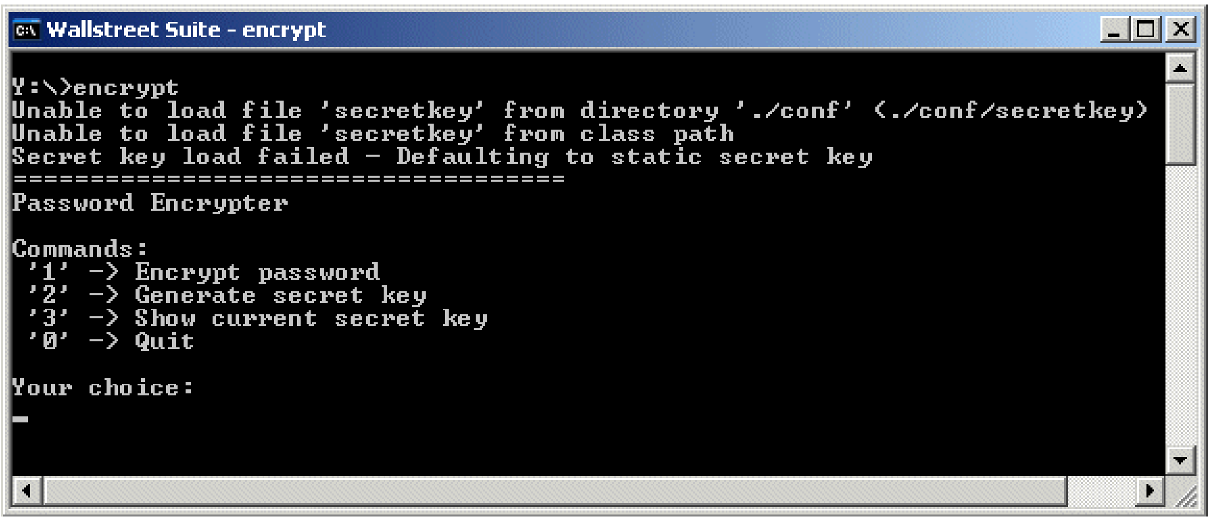
Windows

```
%FK_HOME%\bin\encrypt.bat
```

Unix

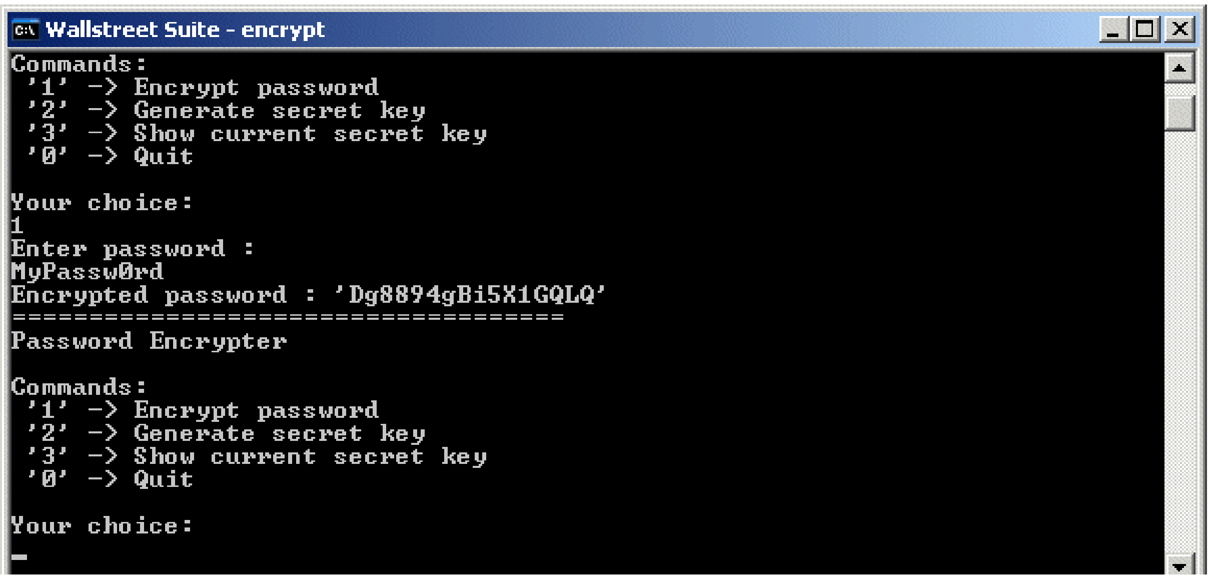
```
$FK_HOME/etc/rc.encrypt
```

This displays a menu (Windows version shown below) where you select the relevant item:



```
Y:\>encrypt
Unable to load file 'secretkey' from directory './conf' (./conf/secretkey)
Unable to load file 'secretkey' from class path
Secret key load failed - Defaulting to static secret key
====================================
Password Encrypter

Commands:
 '1' -> Encrypt password
 '2' -> Generate secret key
 '3' -> Show current secret key
 '0' -> Quit

Your choice:
_
```

For example, to encrypt a password, enter 1 and press the Enter key:



```
Commands:
 '1' -> Encrypt password
 '2' -> Generate secret key
 '3' -> Show current secret key
 '0' -> Quit

Your choice:
1
Enter password :
MyPassw0rd
Encrypted password : 'Dg8894gBi5X1GQLQ'
====================================
Password Encrypter

Commands:
 '1' -> Encrypt password
 '2' -> Generate secret key
 '3' -> Show current secret key
 '0' -> Quit

Your choice:
_
```

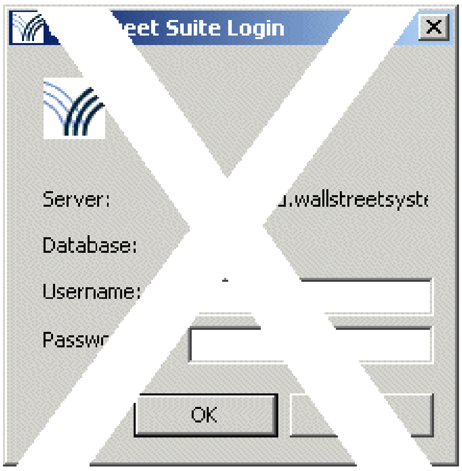
Appendix I Trusted database login from TRM

I.1 Introduction

To start TRM, the user is asked for a login and password to the database when starting Application Manager. Under certain conditions, it is possible to configure a "trusted" connection so that the database login accepts the user’s Windows login and password. The conditions are:

- This is available for an MSSQL or Oracle database only.
- This works for TRM and ACM GUI applications, not for WebSuite, pricing tools, realtime processes, or for technical accounts.

The result is that the standard login dialog is no longer displayed:



I.2 Trusted connection setup

This involves two configuration steps: activation in the client configuration, and database configuration changes in either MSSQL or Oracle.

I.2.1 Client configuration

In the client configuration, the environment variable `FK_TRUSTED_CONNECTION` must be set in order to activate a trusted connection, like this:

```
Set FK_TRUSTED_CONNECTION = 1
```

Two batch files are available for starting up Application Manager and connecting to the WSS database with this environment variable already declared. There is one for trusted users:

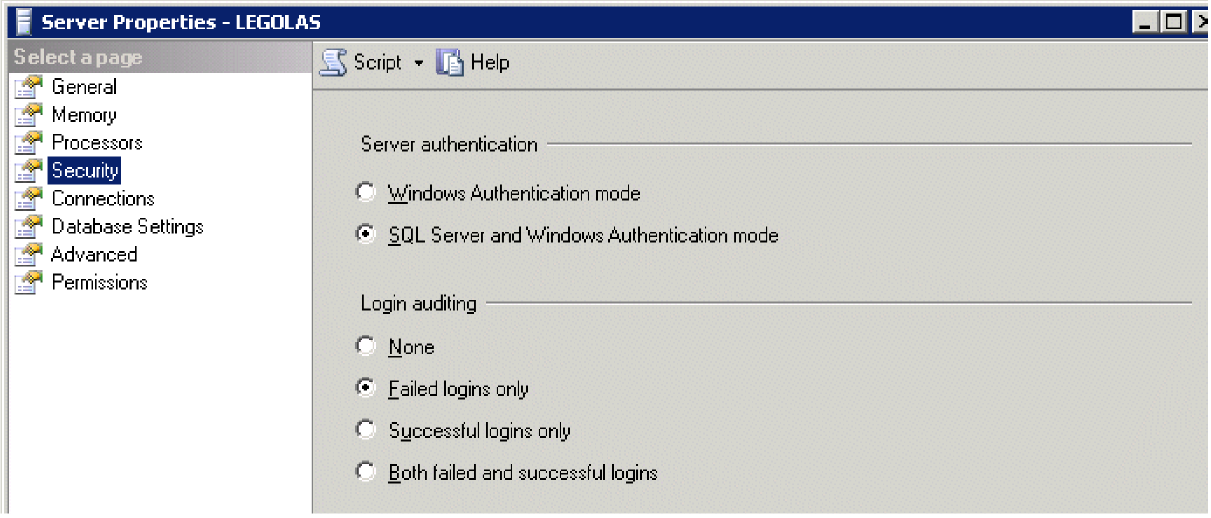
- `%WSS_HOME%\envs\<`*`environment`*`>\bin\trm-trusted-<`*`environment`*`>.bat`

and one for users requiring further authentication:

- `%WSS_HOME%\envs\<`*`environment`*`>\bin\trm-<`*`environment`*`>.bat`

I.2.2 MSSQL database configuration

Open the MSSQL Server properties, select **Security**, and ensure that the **SQL Server and Windows Authentication mode** option is selected:



Server authentication

○ Windows Authentication mode

● SQL Server and Windows Authentication mode

Login auditing

○ None

● Failed logins only

○ Successful logins only

○ Both failed and successful logins

I.2.3 Oracle database configuration

Trusted users are created as externally authenticated accounts. To do this, use the Oracle OS authentication feature. You should set two configuration parameters:

Name	Type	Default value	Required value
`os_authent_prefix`	string	`ops$`	`""`
`remote_os_authent`	Boolean	`False`	`True`

Wallstreet Suite DBCA templates include both these parameters, but they are defaulted to deny a trusted login, as shown in the table above.

Once the values of these parameters have been changed, you must restart the database.

I.2.3.1 Security considerations

Setting `remote_os_authent` to `True` can cause a security exposure, because it allows someone using a non-secure protocol, such as TCP, to perform an operating system-authorized login.

The Wallstreet Suite TRM application ensures that only a user that is authenticated through Wallstreet Suite domain permissions can access the database as a trusted user. So an employee who connects a computer to the network, creates a named account on it, and pretends to be a trusted Wallstreet Suite user using a fake local named account will not succeed.

Here are some recommendations when setting `remote_os_authent` to `True`:

- Only low level user accounts should be created as "trusted users". Administrator accounts, technical accounts or users with high permissions must remain as standard database authenticated acounts.
- Set up a "white list" of computers that are allowed to connect to the database. At customer sites, if users connect to a Citrix server to start the WSS client application, this list is very small. This is an `sqlnet` configuration task on the Oracle server side. For example, add these rows to `sqlnet.ora`, then restart the Oracle Listener:

  ```
  tcp.validnode_checking = yes

  tcp.invited_nodes = (localhost,
  trmserver.mydomain.com,citrixserver.mydomain.com,webserver.mydomain.com)
  ```

Windows machines listed inside this white list must be secured, with restricted administrator permissions.

I.2.3.2 Creating trusted accounts in Security Center

To do this:

1. Open Security Center.
2. Open User Administration Editor.
3. Select the user.
4. Switch on **Trusted User**.

   This will automatically disable the following fields:

   – **Password** and **Retype password**

   – **Max failed Logins**

   – **Password Expiry**

   – **Web User Group**. (Trusted users cannot use the Websuite application, see *I.1 Introduction* on page 161.)

Note: On Oracle, the domain prefix should **not** be used in the **User ID** field. This prefix is mandatory only for MSSQL.

Appendix J WebSuite with an IIS proxy server

This section described how to configure a Microsoft IIS proxy server to work with a WebSuite/WebLogic installation

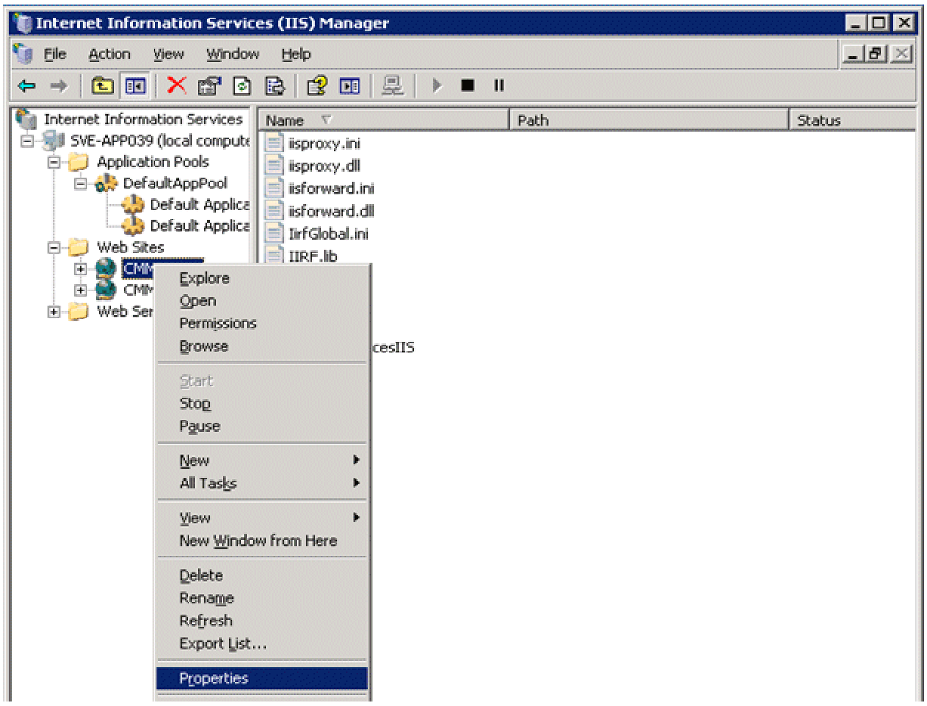
Important: Microsoft IIS must already be installed on the system and the website already created.

J.1 Configuring an IIS proxy with the ISAPI plugin and Weblogic

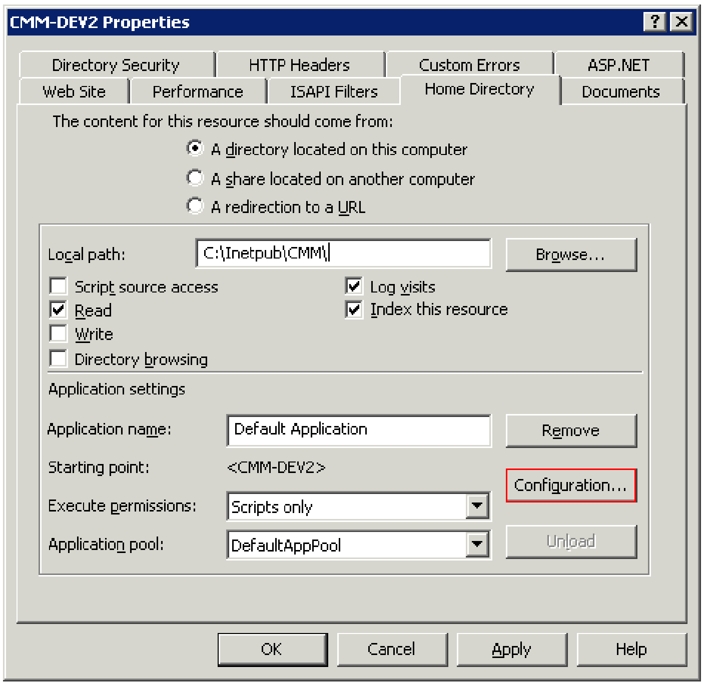
J.1.1 Installing and configuring the ISAPI plugin by proxying by path

Hint: To configure proxying by MIME type too, see: `http://download.oracle.com/docs/cd/E12840_01/wls/docs103/plugins/isapi.html`.

1. Copy the files `iisproxy.dll` and `iisforward.dll` from `${PATH_TO_WEBLOGIC}/server/plugin/win/32` or `${PATH_TO_WEBLOGIC}/server/plugin/win/64` in your WebLogic Server installation to your website folder.
2. Create two files called `iisproxy.ini` and `iisforward.ini`.
3. In Windows Explorer, display your website location, right-click the folder, and select **Properties**.



4. Select the **Home Directory** tab, and click **Configuration** in the **Application settings** section.



CMM-DEV2 Properties

The content for this resource should come from:

● A directory located on this computer
○ A share located on another computer
○ A redirection to a URL

Local path: C:\Inetpub\CMM\

□ Script source access
☑ Read
□ Write
□ Directory browsing
☑ Log visits
☑ Index this resource

Application settings

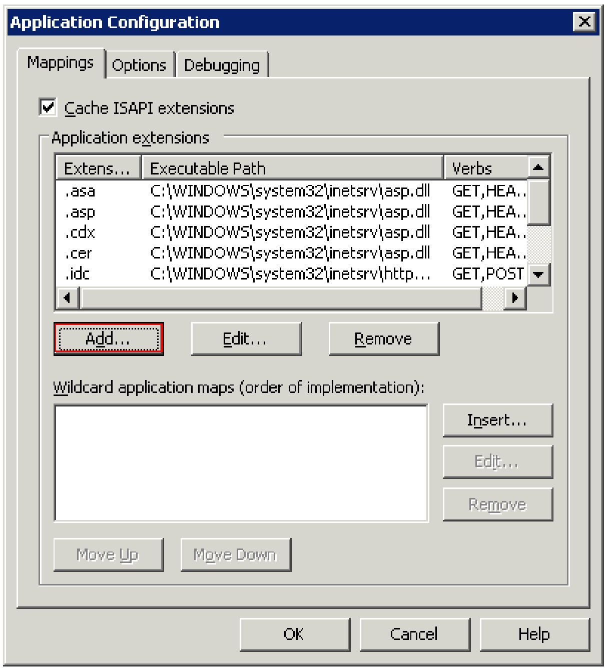
Application name: Default Application

Starting point: <CMM-DEV2>

Execute permissions: Scripts only

Application pool: DefaultAppPool

5. On the **Mappings** tab, click **Add** to add file types and configure them to be proxied to the WebLogic Server.



Application Configuration

☑ Cache ISAPI extensions

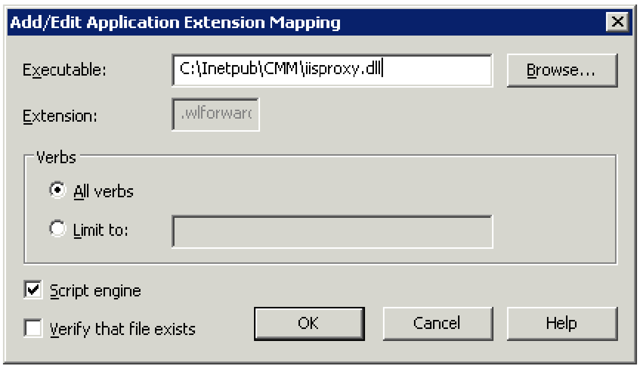
Application extensions

Extens...	Executable Path	Verbs
.asa	C:\WINDOWS\system32\inetsrv\asp.dll	GET,HEA..
.asp	C:\WINDOWS\system32\inetsrv\asp.dll	GET,HEA..
.cdx	C:\WINDOWS\system32\inetsrv\asp.dll	GET,HEA..
.cer	C:\WINDOWS\system32\inetsrv\asp.dll	GET,HEA..
.idc	C:\WINDOWS\system32\inetsrv\http...	GET,POST

Wildcard application maps (order of implementation):

a. In the **Extension** box, enter `.wlforward`.

b. In the **Executable** field, point to the `iisproxy.dll` file.



Add/Edit Application Extension Mapping

Executable: C:\Inetpub\CMM\iisproxy.dll

Extension: .wlforwarc

Verbs

● All verbs

○ Limit to:

☑ Script engine

☐ Verify that file exists

6. Add those lines in `iisproxy.ini` to forward requests to WebLogic:

```
WebLogicHost=WeblogicHostIP

WebLogicPort=20201

WlForwardPath=/

HungServerRecoverSecs=86400

Idempotent=OFF

Debug=ALL
```

7. Add those lines in iisforward.ini to configure the requests that will be sent to WebLogic:

```
vhost1=websiteName:Port

websiteName:Port=C:\Inetpub\CMM\iisproxy.ini
```

J.2 Adding a filter plugin to manage WebSuite local resources

The ISAPI plugin can proxy resources by path or extension but cannot test the existence of a file in a specific path nor rewrite URLs. In WebSuite, resources can be available under several paths, and IIS has to test for the existence of a file.

So a new filter must be added: the Ionics Isapi Rewrite Filter (IIRF).

J.2.1 Installing IIRF

1. Download the latest version of IIRF from `http://www.codeplex.com/IIRF`.

   If, for example, the IIS proxy is set up in `C:\Inetpub\CMM`, you should normally already have four files in this folder:

   ```
   iisforward.dll

   iisforward.ini

   iisproxy.dll

   iisproxy.ini
   ```

2. Extract the archive and copy these files to `C:\Inetpub\CMM`:

   ```
   IIRF.dll

   IIRF.lib
   ```

3. Create two files named:

   ```
   IirfGlobal.ini

   IIRF.ini
   ```

4. In `IirfGlobal.ini`, write this line in it to put the same priority as the ISAPI plugin:

   ```
   RewriteFilterPriority HIGH
   ```

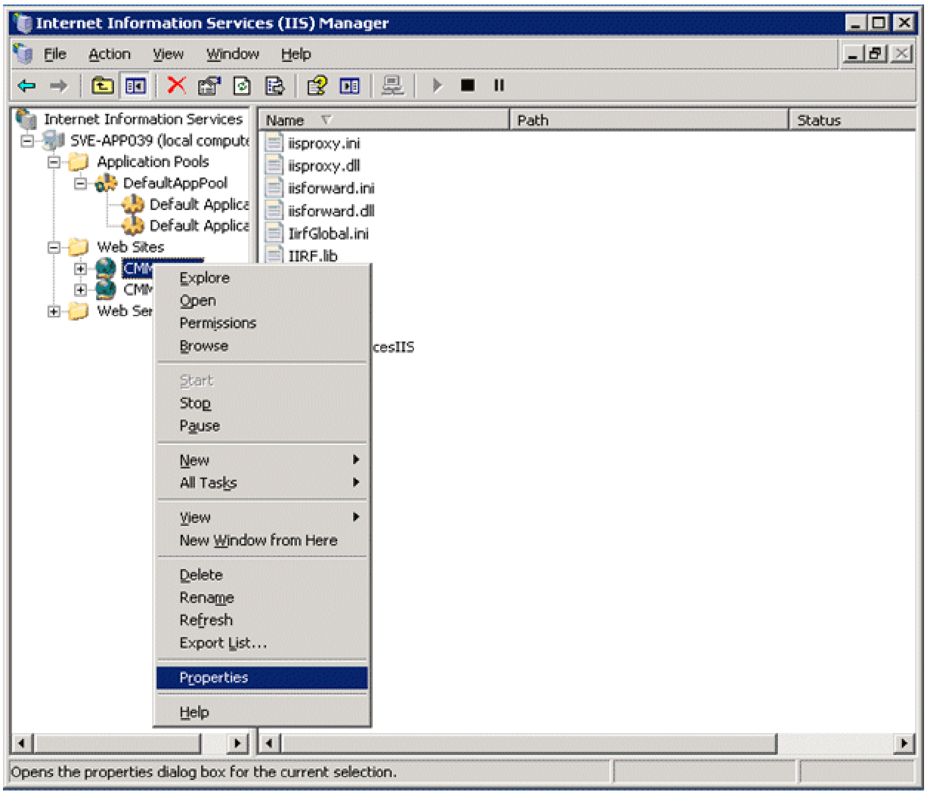
5. In `IIRF.ini`, add the rules that will determine whether or not IIS will handle the display of the resource. For WebSuite, use the rules written below:

   ```
   #remove # comments if you want to have logfiles
   #RewriteLogLevel 3
   #RewriteLog c:\temp\Iirf
   #StatusUrl  iirfstatus

   #test if the file exists with a simple URI (like /websuite/image/test.gif)
   RewriteCond c:/Inetpub/CMM/cmm-dev2/$1 -f
   RewriteRule ^/(.*).wlforward$ /$1 [L]

   #test if the file exists with a modify URI that points to custom style (like
   /websuite/style/custom/image/test.gif)

   RewriteCond c:/Inetpub/CMM/cmm-dev2/websuite/style/custom/$1 -f
   RewriteRule ^/websuite/(.*).wlforward$ /websuite/style/custom/$1 [L]

   #test if the file exists with a modify URI that points to silver style (like
   /websuite/style/silver/image/test.gif)
   RewriteCond c:/Inetpub/CMM/cmm-dev2/websuite/style/silver/$1 -f
   RewriteRule ^/websuite/(.*).wlforward$ /websuite/style/silver/$1 [L]

   #test if the file exists with a modify URI that points to classic style (like
   /websuite/style/classic/image/test.gif)
   RewriteCond c:/Inetpub/CMM/cmm-dev2/websuite/style/classic/$1 -f
   RewriteRule ^/websuite/(.*).wlforward$ /websuite/style/classic/$1 [L]

   #if all tests return false, then weblogic handles the request.
   ```

6. Add the IIRF filter in IIS:

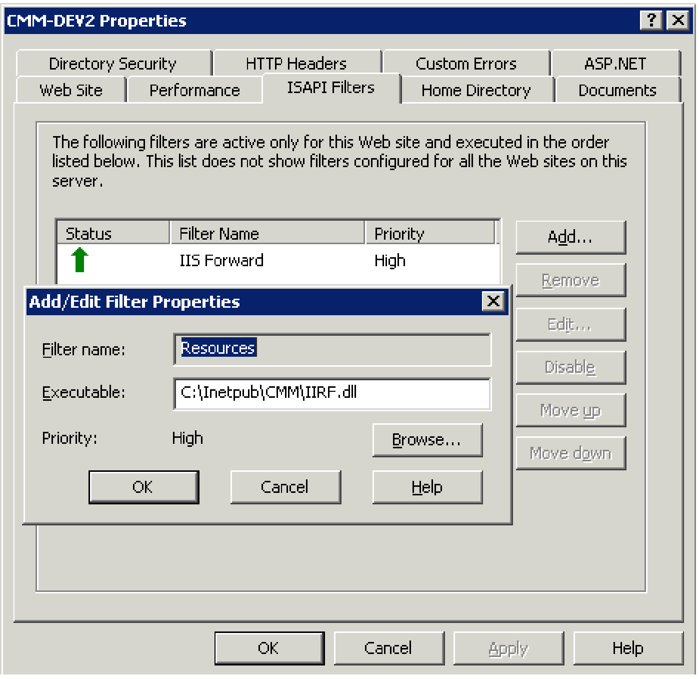
   a. Open IIS Manager.

b. Right-click on your website, and select **Properties**.



Opens the properties dialog box for the current selection.

c. Open the tab **ISAPI Filters** and and click **Add** to add the `IIRF.dll` file. In the screenshot below, the filter has been called `Resources`.



The following filters are active only for this Web site and executed in the order listed below. This list does not show filters configured for all the Web sites on this server.

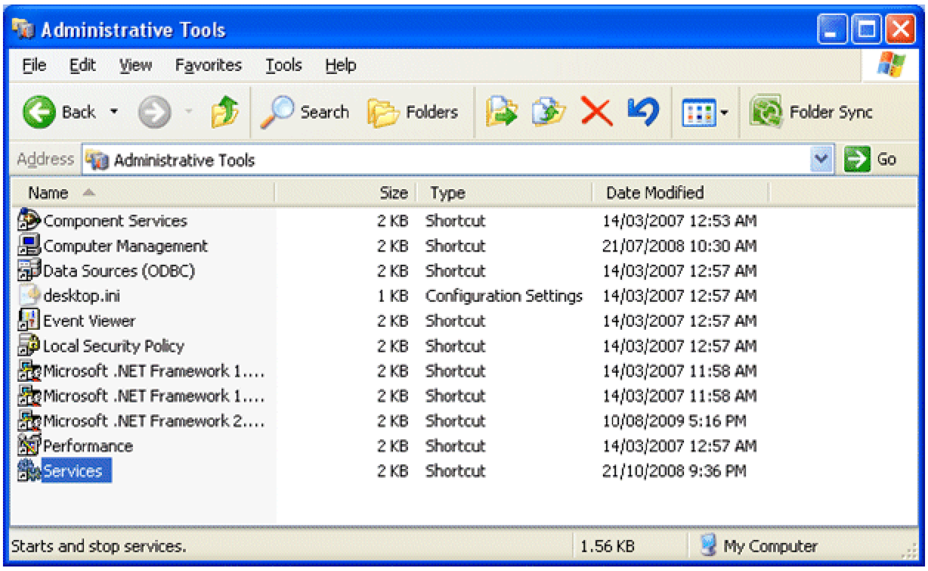
Status	Filter Name	Priority
⬆	IIS Forward	High

Filter name: Resources

Executable: C:\Inetpub\CMM\IIRF.dll

Priority: High

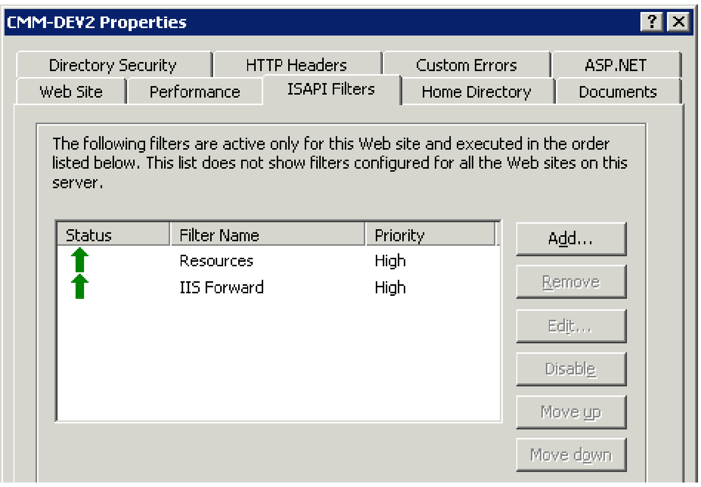
d. Restart IIS (**Services** in the Administrative Tools in the Control Panel)



Name	Size	Type	Date Modified
Component Services	2 KB	Shortcut	14/03/2007 12:53 AM
Computer Management	2 KB	Shortcut	21/07/2008 10:30 AM
Data Sources (ODBC)	2 KB	Shortcut	14/03/2007 12:57 AM
desktop.ini	1 KB	Configuration Settings	14/03/2007 12:57 AM
Event Viewer	2 KB	Shortcut	14/03/2007 12:57 AM
Local Security Policy	2 KB	Shortcut	14/03/2007 12:57 AM
Microsoft .NET Framework 1....	2 KB	Shortcut	14/03/2007 11:58 AM
Microsoft .NET Framework 1....	2 KB	Shortcut	14/03/2007 11:58 AM
Microsoft .NET Framework 2....	2 KB	Shortcut	10/08/2009 5:16 PM
Performance	2 KB	Shortcut	14/03/2007 12:57 AM
Services	2 KB	Shortcut	21/10/2008 9:36 PM

Starts and stop services. 1.56 KB My Computer

e. Verify that the plugin is installed correctly (green arrow), and put the IIRF plugin at the top of the list.



The following filters are active only for this Web site and executed in the order listed below. This list does not show filters configured for all the Web sites on this server.

Status	Filter Name	Priority
↑	Resources	High
↑	IIS Forward	High

f. Copy the resources under IIS.

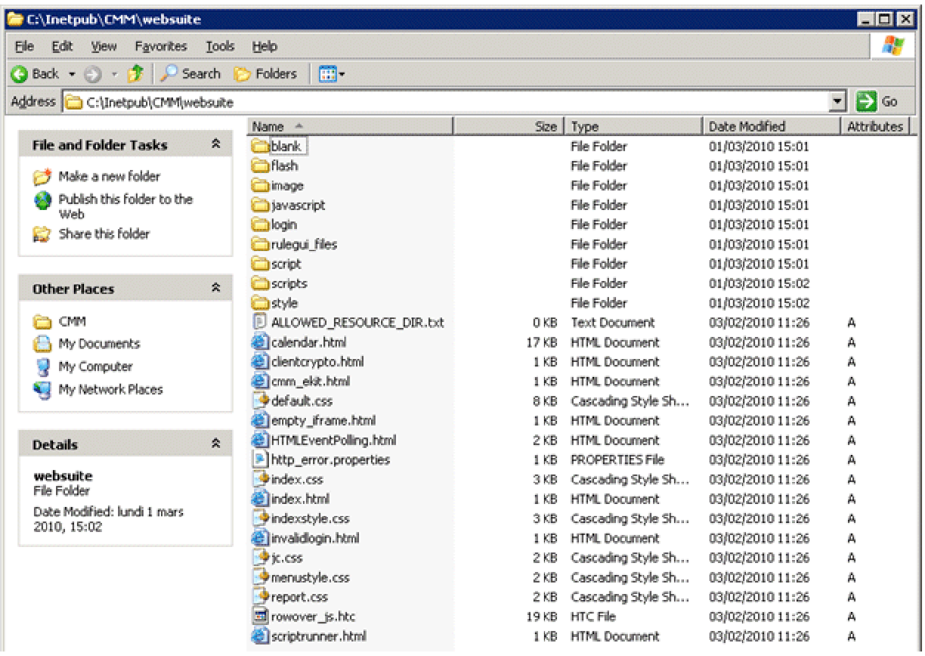
Copy the `ws-suite\envs\${env}\etc\wss-web\websuite` folder to `C:\Inetpub\CMM\` but do not copy the following:

```
WEB-INF
```

```
META-INF
```

```
xml_handler_spec
```

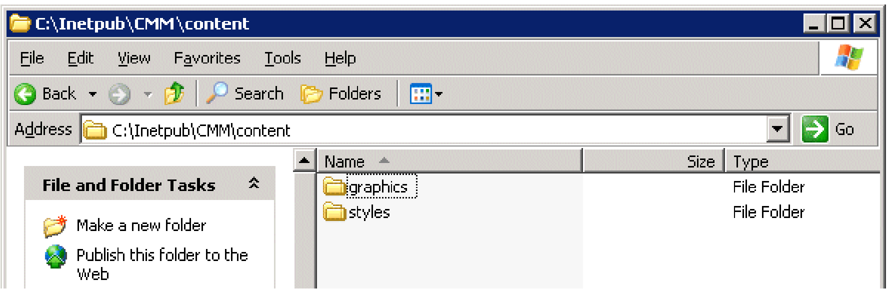
```
xsl
```



websuite
File Folder
Date Modified: lundi 1 mars 2010, 15:02

Name	Size	Type	Date Modified	Attributes
blank		File Folder	01/03/2010 15:01	
flash		File Folder	01/03/2010 15:01	
image		File Folder	01/03/2010 15:01	
javascript		File Folder	01/03/2010 15:01	
login		File Folder	01/03/2010 15:01	
rulegui_files		File Folder	01/03/2010 15:01	
script		File Folder	01/03/2010 15:01	
scripts		File Folder	01/03/2010 15:02	
style		File Folder	01/03/2010 15:02	
ALLOWED_RESOURCE_DIR.txt	0 KB	Text Document	03/02/2010 11:26	A
calendar.html	17 KB	HTML Document	03/02/2010 11:26	A
clientcrypto.html	1 KB	HTML Document	03/02/2010 11:26	A
cmm_ekit.html	1 KB	HTML Document	03/02/2010 11:26	A
default.css	8 KB	Cascading Style Sh...	03/02/2010 11:26	A
empty_iframe.html	1 KB	HTML Document	03/02/2010 11:26	A
HTMLEventPolling.html	2 KB	HTML Document	03/02/2010 11:26	A
http_error.properties	1 KB	PROPERTIES File	03/02/2010 11:26	A
index.css	3 KB	Cascading Style Sh...	03/02/2010 11:26	A
index.html	1 KB	HTML Document	03/02/2010 11:26	A
indexstyle.css	3 KB	Cascading Style Sh...	03/02/2010 11:26	A
invalidlogin.html	1 KB	HTML Document	03/02/2010 11:26	A
jc.css	2 KB	Cascading Style Sh...	03/02/2010 11:26	A
menustyle.css	2 KB	Cascading Style Sh...	03/02/2010 11:26	A
report.css	2 KB	Cascading Style Sh...	03/02/2010 11:26	A
rowover_js.htc	19 KB	HTC File	03/02/2010 11:26	A
scriptrunner.html	1 KB	HTML Document	03/02/2010 11:26	A

7. Create the folder `C:\Inetpub\CMM\content` and in the `ws-suite\envs\${env}\etc\wss-web\cmm\VirtualDirectory\` path, copy the `graphics` and `styles` folders:



Name	Size	Type
graphics		File Folder
styles		File Folder

Make sure that the `websuite` and `contents` folders allow users (and therefore IIS) to read files.

8. To ensure that the setup is correct, verify that access logs are enabled in WebLogic:

   a. In the left pane of the Console, expand **Environment** and select **Servers**.

   b. In the **Servers** table, click the name of the server instance whose HTTP logging you want to configure.

   c. On the **Logging: HTTP** page, encure that the **HTTP Access Log File Enabled** checkbox is checked.

Hint:

For more information:
`http://download.oracle.com/docs/cd/E13222_01/wls/docs91/ConsoleHelp/taskhelp/logging/EnableAndConfigureHTTPLogs.html`

Appendix K Troubleshooting and FAQs

K.1 Troubleshooting

K.1.1 Common Errors

The database creation scripts are failing

- For Oracle, is the TNS name entry created in the Oracle client `tnsnames.ora` file?
- For Oracle did you use the template to create the DB Instance?
- Does the DB instance have all the tablespaces defined in the templates?
- For MSSQL, did you create the data files on the MSSQL server?

K.1.2 Suite Installer troubleshooting

First examine the installation log files.

If the Wizard-style Installer wss used, also check the `TSI/ant.install.log` file.

The logging level of the `si.log` can be tuned through the file `SI_Manager\packagemanager\si\logging.ini`

K.1.3 Server process troubleshooting

To control server logging, see *6.6 Logging* on page 77.

K.1.3.1 Log in console

On Windows, it is possible to get a console for each process launched.

To enable the console display on Windows, edit the `sharedconf/components/pmm/config.properties` file and set the `pmm.debug.param` parameter to `--debug`. Then run a `reconfigure`.

Alternatively, set `PMM_DEBUG=--debug` in your environment.

K.1.3.2 Run any server from the command line

K.1.3.2.1 To run servers from the command line

1. Open the PMM configuration file `pmm/etc/conf/env_<environment>.xml` and look for the process to debug.

   For example, to start the TRM - Settlement serviced on the command line, go to the section:

   ```
   <process id="trm_settlement_serviced" desc="      TRM - Settlement serviced" ...>
   ```

2. Open a shell.
3. Set the environment variables that would be set by PMM. For example, with settlementd:

   ```
   <set var="FK_COMKIT_SETUP" value="settlement" />

   <set var="TS_LOG_FILE"
   value="E:\intallation\envs\dev\var\logs\trm\serviced_settlement.log" />
   ```

```
<set var="SERVICED_CONFIG_FILE"
value="E:\intallation\components\trm\etc\serviced\settlement.config" />

<set var="FK_LOGIN" value="[$(FK_SERVICED_LOGIN)]" export="true"/>

<set var="PMM_PID_FILE"
value="E:\intallation\envs\dev\var\pid\settlementserviced.pid" />
```

Note: `FK_SERVICED_LOGIN` is defined inside `pmm\etc\conf\private\env_<environment>.secret`.

4. Set debug variables if required (see *6.6 Logging* on page 77).
5. Make sure that `PMM_DEBUG=--debug`
6. Execute the `init.bat`/`sh` command path followed by the argument(s). For example:

   ```
   <command path="E:\intallation\envs\dev\bin\services\init.bat"
   arguments="serviced" />
   ```

   Execute :

   ```
   >E:\intallation\envs\dev\bin\services\init.bat serviced
   ```

K.1.3.2.2 Process command lines

Process command lines are available inside the `envs\dev\var\logs\pmm` directory (last line of each log file). This can be useful for determining the startup options.

To change startup options, first look at the process PMM command section:

```
<command path="<init.bat/sh path>" arguments="<service> [py]" />
```

- If the service element is specified with `py`, the `init.bat`/`sh` script will call the <service>`.py` python script inside the `FK_HOME\bin` directory. Edit this file and add your extra parameters.
- If the service element is specified without `py`, the `init.bat`/`sh` will load the startup command and options from the `envs\<environment>\bin\services\<service>.bat`/`sh` file. Edit this file and add your extra parameters.

K.1.3.3 Customizing any server using PMM

Server launch customizing can be performed using PMM templates. See *6.3.1.2 Customizing real-time services from the site directory* on page 75.

K.1.3.3.1 Setting environment variables per process in PMM

Setting non global environment variables that apply so only for a server process is possible either through PMM configuration or through the environment.

Through PMM configuration

In PMM, customize the server configuration by adding extra set statements:

```
<set var="<ENV_VARIABLE>" value="<value>" export="true"/>
```

It is recommended that you handle it through PMM templates. See *6.3.1.2 Customizing real-time services from the site directory* on page 75.

Through the environment

This solution is based on `TS_SERVICE_NAME` environment variable testing.

An example is shown in `envs\<environment>\etc\environment\parts\99_debug_realtimes.bat`

K.1.4 Tomcat application server and java applications troubleshooting on Windows

On Windows, Tomcat application servers and java applications like TRMSWift and DMM can be started without the console. To enable the console display on Windows, edit the `sharedconf/environment.properties` file and set the `ts.java.windows.exec.javaw` parameter to `false`. Then run a `reconfigure`.

K.1.5 Weblogic application server troubleshooting

Weblogic logging: see *6.6.2 Log configuration* on page 78.

Under Weblogic, application servers can be monitored through the Admin Server console.

Open a web browser and connect to `http://${hostname}:${ts.env.portbase}51/console`.

Note: All Weblogic servers can be started through the command line using start/stop scripts under directory `envs\<environment>\bin\weblogic`

K.1.6 ACM Application Server Troubleshooting

When you have completed the ACM Application Server installation, you can quickly check that the server is available by using your web browser. First you have to find out what is the URL the server is accessible at. The URL is composed from the host name and the port where ACM Application server is running. It is also necessary to provide the correct context - the correct value is specified in `ACM_WEB_CONTEXT` environment variable and may differ for each application server platform. If you already have a client installation you can easily find out the URL in the following way:

1. Open a command line window with evaluated environment.
2. Run `echo http://%ACM_HTTP_HOST%:%ACM_HTTP_PORT%/%ACM_WEB_CONTEXT%`

The output of the command represents the URL you have to enter into your web browser. An info page with the ACM package signature should appear. If it does not, then it means that ACM Application Server startup was not successful and a critical error occured that prevented ACM Application Server from initializing at the level of the container server. If the info page appears, then the ACM Application Server has started - this however does not mean that all necessary processes are running.

If you see the info page but still experience problems with ACM, examine the ACM Application Server log files. The directory for the ACM log files is determined by the value in `ACM_LOG` environment variable. The logs contain useful information about the startup parameters, database connection strings as well as exception that might have occured during the ACM Application Server startup.

One of the most common problems is a misconfigured database connection - please check the value of the `ACM_JDBC_URL` variable (in both environment and the log file) and verify that the database server specified in this variable exists, and that it is up and running.

Another common error is represented by the following text in ACM log file:

Error creating bean with name 'RuntimeProperties'...

This usually signals that the property file containing the environment snapshot was not created and/or it cannot be read by ACM Application Server. The means for specifying the property file location and also the means for creating the file may differ for each application server platform.

K.2 FAQs

FAQ

Can I remove administrator permissions on Windows machines?

Administrator permissions must be set during installation (see *5.1 Prerequisites* on page 57) and should not be removed on the server side.

If the installation is performed on a remote server, but TRM client applications are started on end-user PCs, at least one operation must be done locally that requires temporary administrator's permissions:

1. if a Windows 32bit package has been installed, in a TRM-evaluated shell, run this command with administrator permissions:

   ```
   regsvr32 %FK_HOME%\bin\vcfi32.ocx

   regsvr32 %FK_HOME%\bin\TeeChart8.ocx

   regsvr32 %FK_HOME%\bin\Calendar.ocx

   %FK_HOME%\bin\vcredist.bat
   ```

2. Change the privileges back to normal user permissions.

   TRM will now run on the end-user's PC with these permissions.

FAQ

What WSS elements can I remove so that I can have a thin client?

With the existing version of Suite Installer, to install a client, you must perform a complete installation. If you need to reduce the size of the client installation, you can remove the following directories:

- `%FK_HOME%\share\mssql`
- `%FK_HOME%\share\sybase`
- `%FK_HOME%\share\oracle`
- `%FK_HOME%\share\common`
- `%FK_HOME%\etc\trmswift`

FAQ

How do I override a property file for a specific environment?

By placing a copy of the default properties file from `..\sharedconf` in `..\envs\<wss environment>\conf\database.properties`.

FAQ

How do I change default environment parameters for TRM?

Either

- Find the correct property in `sharedconf`, of `../conf`.

Or

- Place a environment script in `..\envs\`<environment>`\etc\environments\parts`. The scripts are executed in the order listed in the directory. This means that a script called `90_sitechanges` will be executed last and properties set here will override all other properties.

FAQ

Where are the database connections defined?

See *Appendix A Configuration Properties* on page 125

But in general in database.properties in either:

- ..\sharedconf\database.properties

Or

- ..\envs\<wss environment>\conf\database.properties

FAQ

How do I get the environment to re-process the property files once I have made a change?

See "Reconfiguring the installation.

FAQ

Why do files that are deployed from the site get overridden by the Installer?

If the site command is run with the -w option (reconfigure) or if a reconfigure command is run, the Installer deploys scripts and environment files that will override existing ones in the installation. To make sure that all site files are applied in the installation, it is recommended that you run the site command afterwards to redeploy the site files.

FAQ

How do site folders work with the Wallstreet Suite Package Manager?

The site folder is untouched by the Wallstreet Suite Package Manager, this allows the environments to be managed using the normal site folder ways.

Some items to include in the site folder could be: database.properties, 3rdpartysoftware.properties, host.properties, etc.

FAQ

On Windows, why are Tomcat application servers closed when I log off from the current session?

This is because java applications interpret Windows LOGOFF events as shutdown events.

To disable this behavior, edit the sharedconf/environment.properties file and set the ts.java.runtime.option.xrs parameter to true. Then run a reconfigure.

FAQ

Why is there one omniNames service per environment?

In order to provide isolation between environments, each environment has its own naming service. This allows you to start and stop complete environments without impacting other environments using the same binaries.

FAQ

Is there a sample CSD?

Yes, a sample CSD is located in \SI_Manager\suite-packages\templates\simple-extract-package

Each CSD includes step by step instructions on how to create new ones based on the template.

FAQ

How can I move log files or import/export files to a particular directory?

This can be done manually on a Unix platform using symbolic links.

Using symbolic links for CMM import/export

1. Create your destination directory:

```
mkdir <mydir>/<env>/CMM/ImportExport

mkdir <mydir>/<env>/CMM/Audit
```

2. Create symbolic links:

```
ln <directory_structure>/envs/<env>/etc/wss-web/c,,/Runtime/var/
ln -s <WSSInstallation_dir>/envs/<env>/CMM/ImportExport ImportExport

ln -s < WSSInstallation_dir >/ envs/<env>/CMM/Audit Audit
```

Using symbolic links for logs

1. Create your destination directory:

   ```
   mkdir <mydir>/WSS/logs/<env>/cmm_logs

   mkdir <mydir>/WSS/logs/<env>/cmm_var

   mkdir <mydir>/WSS/logs/<env>/env_logs
   ```

2. Create symbolic links:

   For each directory remove the the existing log and directory, and create the adequate symbolic link in:

   ```
   ln -s <mydir>/WSS/logs/<env>/env_logs  <WSSInstallation_dir>/ envs/<env>/var/logs

   ln -s <mydir>/WSS/logs/<env>/cmm_logs  <WSSInstallation_dir>/
   envs/<env>/etc/wss-web/cmm/VirtualDirectory/logs/cmm

   ln -s <mydir>/WSS/logs/<env>/cmm_var  <WSSInstallation_dir>/
   envs/<env>/etc/wss-web/cmm/Runtime/var
   ```

FAQ

How can I specify a particular timezone in the application server?

You can specify a particular timezone by modifying the `appservers.properties` file in your `site` structure:

```
ts.appserver.timezone=GMT
```

For exanmple, in an existing installation, you can modify the following file

```
<WSSInstallation_dir>/ envs/<env>/bin/weblogic/start_cmm-weblogic.sh,
```

by replacing `timezone=GMT` by `timezone=Europe/Paris`

FAQ

How can I configure a Corba application behind a firewall?

For some installations that have particular network policies, the Wallstreet Suite Corba processes must be configured behind a firewall.

1. orb.conf.

   You must modify the `orb.conf` file from your `site` environment, or directly in `<WSSInstallation_dir>/ envs/<env>/etc/orb.conf`

   ```
   InitRef = NameService=corbaname::1.2@10.35.0.131:20200

   #endPoint = giop:unix:
   #     = giop:tcp::
   #     = giop:ssl::

   clientTransportRule = 10.35.0.131       bidir,unix,tcp
                 = *                       ssl,tcp,bidir

   serverTransportRule = 10.35.0.131          bidir,unix,tcp,ssl
                 = *                          bidir,tcp,ssl
   ```

   This must also be done on the client.

2. You must also add a firewall option to launch processes.

3. You must modify the `20_trm.pl` file in your `site` environment, or directly in `<WSSInstallation_dir>/ envs/<env>/etc/environment/parts/20_trm.pl`:

```
$ENV{FK_MDSD_ARGS}="-ORBendPoint giop:tcp::20220";

$ENV{FK_LIMITD_SETUP}="--portfolio-id FINANCIAL-FLOWS --contexts 1 --state-id
OPEN --scenario-id FREEZE --interval 3600000 --min-log-interval 43200000
--log-interval 43200000 -ORBendPoint giop:tcp::20220";
```

FAQ

How does Suite Installer manage and attribute port numbers?

Suite Installer uses the environment number (specified in the **Environment number** field in Suite Installer) to attribute all port numbers.

A three-digit prefix number is created as follows:

The environment number **N** which can be between 1 and 99, and a variable defined in the `environment.properties` file (`rc.env.portbase` with default value equal to `2 ${ts.env.portnumber}`):

Example:

If environment number = 1, the prefix will be 201

If environment number = 2, the prefix will be 202

After, this prefix is used to allocate port numbers as follows:

Omniorb

Some properties are used to create `orb.conf` configuration files.

These properties can be overridden in the `environment.properties` file.

Property	Value	Default value
`nameserver.port`	<prefix>00	`20100`

Active MQ

Some properties are used to create `active.xml` configuration files.

These properties can be overridden in the `message-bus.properties` file.

Property	Value	Default value with n = 1
`activemq.transport.openwire.port`	<prefix>60	`20160`
`activemq.transport.ssl.port`	<prefix>61	`20161`
`activemq.transport.stomp.port`	<prefix>62	`20162`
`activemq.transport.default.port`	<prefix>63	`20163`
`activemq.management.jmx.port`	<prefix>64	`20164`

WebSuite (TRMWeb)

Some properties are defined as environment variables. These properties are generated during installation.

Property	Value	Default value with n = 1
`EKIT_MESSAGE_PORT`	<prefix>51	`20151`
`EKIT_JNDI_SERVER_PORT`	<prefix>52	`20152`
`EKIT_JNDI_SERVER_RMI_PORT`	<prefix>53	`20153`

Web server

Some properties are used to define port numbers in WebSuite.

(These properties can be overridden in the `appservers.properties` file.

Property	Value	Default value with n = 1
`ts.appserver.t1.http.port`	<prefix>01	20101
`ts.appserver.t1.shutdown.port`	<prefix>02	20102
`ts.appserver.t1.https.port`	<prefix>03	20103
`ts.appserver.t2.http.port`	<prefix>04	20104
`ts.appserver.t2.shutdown.port`	<prefix>05	20105
`ts.appserver.t2.https.port`	<prefix>06	20106
`ts.appserver.t3.http.port`	<prefix>07	20107
`ts.appserver.t3.shutdown.port`	<prefix>08	20108
`ts.appserver.t3.https.port`	<prefix>09	20109
`ts.appserver.t6.http.port`	<prefix>16	20116
`ts.appserver.t6.shutdown.port`	<prefix>17	20117
`ts.appserver.t6.https.port`	<prefix>15	20118

In `weblogic.properties`:

Property	Value	Default value with n = 1
`weblogic.admin.server.port`	<prefix>50	20150
`weblogic.admin.server.https.port`	<prefix>51	20151

In `appservers.tomcat.properties`:

Property	Value	Default value with n = 1
`ts.appserver.t1.tomcat.jmx.port`	<prefix>90	20190
`ts.appserver.t2.tomcat.jmx.port`	<prefix>91	20191
`ts.appserver.t3.tomcat.jmx.port`	<prefix>92	20192

Property	Value	Default value with n = 1
`trm.web.message.port`	<prefix>70	20170
`trm.web.naming.service.jndi.port`	<prefix>71	20171
`trm.web.naming.service.jndi.rmi.port`	<prefix>72	20172

Property	Value	Default value with n = 1
cmm.task.scheduler.port	<prefix>13	20113

FAQ

What are the differences between an HP-UX system and other systems?

For Suite Installer, there are no known differences; it considered as a Unix environment.

However, we have come across the following issues, principally owing to an incorrect installation of the system.

1. Python: `import gzip` not working
   - Verify that the library `zlib` is correctly installed and linked to python. Otherwise install it from `www.zlib.net`.
2. Cannot launch Suite Installer in "Wizard" mode
   - The Wizard mode requires graphical user interface support, so an X Server is required to launch it.

     In the case of a remote connection to the server, an X11 forwarding protocol must be used.

FAQ

Can I launch the "Wizard-style" GUI version of Suite Installer on Unix?

The Suite Installer Wizard is based on Java, and so can be launched on any platform that supports Java. However, the Wizard mode requires graphical user interface support, so an X Server is required to launch it.

In the case of a remote connection to the server, an X11 forwarding protocol must be used.

FAQ

How do I install and configure the WSS suite on separate machines?

With the current version (7.2.2.1), there is no simple way to do this. You need to install all components on all participating machines, then configure Process Monitor (one **pmsd** and several **pmad**s) to reflect the structure that you want.

Example :

You want to have:

- One server running WebSuite, called **serverWeb**.
- One server running the realtime processes, called **serverRuntime**.

Perform a complete installation on each of these machines. Afterwards, you will separate and determine how the processes are running by modifying Process Monitor.

It is decided to launch **pmsd** on serverRuntime, and **pmad** on each machine (serverWeb and serverRuntime).

1. Declare the two agents:

   In `$PMM_HOME/etc/conf/pmsd_config.xml`, create a node for each machine in the `<agents>` block:

   ```
     <agents>
       <agent id="serverWEB" desc="WEB Server">
         <host address="serverWEB.com" port="8818" />
       </agent>
       <agent id="serverRuntime" desc="Factory Server">
         <host address=" serverRuntime .com" port="8818" />
       </agent>
     </agents>
   ```

2. Specify which agent will launch what:

   In `$PMM_HOME/etc/conf/env_<your_env>.xml`, specify which agent is responsible for launching each process (i.e. on which servers the processes will be launched):

   ```
   <process id="cmm_tcmd_appserver"
            desc="       TCMD - Weblogic application server"
   ```

```
         agent="serverWEB">
(…)
<process id="websuite_appserver"
         desc="       WSS Web - Weblogic application server"
         agent="serverWEB">

<process id="sdm_comkit_sdm"
         desc="       SDM - Comkit service "
         agent="serverRuntime">

  <process id="omninames"
         desc="    Naming service"
         agent=" serverRuntime " >
```

Note: Verify that the path specified in "log" , in "pid" or in "command" is correctly defined and exists on the target machine.

3. Launch PMM Admin (Process Monitor) to start and stop processes running on several machines. Your PMM Admin must be connected to the machine where **pmsd** is running.

FAQ

How do I manage files.site?

This describes how to update the installation using `Install.bat`/`sh` and `files.site`.

This is a legacy method of updating the installation with files from the `site` directory. Follow these steps:

1. Create a list of files that have been created and/or modified in the `site` directory, and save it to a file called `files.site`.

2. In the `<installation>/envs/<environment>/etc` directory, run the `install.sh` (Unix) or `Install.bat` (Windows) file, with the following syntax:

`Install` [`-t` *TRM_HOME*] [`-s` *SITE_HOME*] [`-f` *PATH_TO_files.site*] [`-u`]

Where:
TRM_HOME is the path to directory above that of the TRM binaries. By default, this value is `$FK_HOME` (Unix) or `%FK_HOME%` (Windows).
SITE_HOME is the path to the `site` directory. By default, this value is `$WSS_HOME` (Unix) or `%WSS_HOME%` (Windows).
`PATH_TO_files.site` is the path and file name of the `files.site` file.
`-u` is the option for uninstalling.

Under Windows, Install copies the files in `files.site` (or the file that you specified in the command above) into the *TRM_HOME* directory and backs up the original versions.

Under Unix, Install links to the files in `files.site` (or the file that you specified in the command above) from the *TRM_HOME* directory. You should see something like this:

```
#####################################################
Statistics:
Saving away file                                 : 0
Saving away directory                            : 0
No previous version of file                      : 2
Removing old link                                : 776
Linking file/directory                           : 777

Warnings:
Double entry in files.site                       : 189

Errors:
Filename mismatch                                : 1
Cannot create link                               : 1
```

```
    ####################################################
    Time elapsed: 1 seconds.
```

Note: Be careful of conflicts if you are updating the installation from different site directories.

FAQ

How can I add a particular option to comKit with Process Monitor?

With Process Monitor (PMM), you can modify the arguments used to launch comKit by changing the following property: `FK_COMKIT_ARGS` in the file `%PMM_HOME%\etc\conf\env_<my_env>.xml`:

```
<process id="trmswift_comkit_transaction"
         desc="       TRMSwift - Transaction Comkit service "
         agent="factory">
  <application
pidfile="C:\SI_Install\IntegrationTests_NEW\ws-suite\envs\intTest72-1\var\pid\TRMSwiftco
mkitdtransaction.pid" />
  <log
file="C:\SI_Install\IntegrationTests_NEW\ws-suite\envs\intTest72-1\var\logs/pmm/pmm_init
-TRMSwiftcomkitdtransaction.log"/>
  <manage control="AUTOMATIC">
    <start timeout="100000">
      <command
path="C:\SI_Install\IntegrationTests_NEW\ws-suite\envs\intTest72-1\bin\services\init.bat
"
               arguments="comkitd py"/>
      <variables>
        <set var="FK_LOGIN" value="[$(FK_TRMSWIFT_LOGIN)]" export="true"/>
        <set var="FK_COMKIT_SETUP" value="TRMSwift" />
        <set var="FK_COMKIT_SERVICE" value="transaction" />
        <set var="FK_COMKIT_ARGS" value="<MY ARGS>" />
        <set var="TS_LOG_FILE"
value="C:\SI_Install\IntegrationTests_NEW\ws-suite\envs\intTest72-1\var\logs\trmswift\tr
mswift_comkit_transaction.log" />
        <set var="PMM_PID_FILE"
value="C:\SI_Install\IntegrationTests_NEW\ws-suite\envs\intTest72-1\var\pid\TRMSwiftcomk
itdtransaction.pid" />
      </variables>
    </start>
    <dependencies>
      <requires process="trm_mdsd" />
      <requires process="messaging" />
    </dependencies>
  </manage>
</process>
```

This argument will be passed to the `comkitd.py` managed by Process Monitor.

FAQ

How do I script pm_cmd?

With PMM, you can launch processes from the command line, using `pm_cmd` (Unix and Windows platforms). For this to work, PMAD and PMSD must be running. The command syntax is:

`pm_cmd[.exe]` *`user/password@host:port command [args]`*

Some Windows examples:

```
rem Start All Processes
G:\WSsuite72\wssXX\pmm\bin\pm_cmd.exe admin/password@localhost:8037 start
WALLSTREET_SUITE@wsprod3

rem Re-Start All Processes
G:\WSsuite72\wssXX\pmm\bin\pm_cmd.exe admin/password@localhost:8037 restart
WALLSTREET_SUITE@wsprod3
```

FAQ

How do I remove an existing environment?

To remove an existing environment:

- Files:
  - Remove the environment folder located in the `envs` folder.
  - Remove the related files in Process Monitor and restart PMM services.
- Database: drop the database.

FAQ

How do I configure Onyx services?

From version 7.2.2.3, Onyx properties files can be found under *`environment`*`/etc/onyx`. The main properties are set by Suite Installer during the installation phase.

FAQ

Does WSS support Sybase 15 ?

Wallstreet Suite supports Sybase 15 from WSS version 7.2.2.2. This requires changing some `db.version` properties in the `/shared-conf/database.properties` file. See *db.version:* on page 127.

You must already have a Sybase OCS client installed.

in a Windows environment, an error can occur owing to a change in the Sybase OCS structure. You should to copy the following directory:

```
copy -r %SYBASE_HOME%\locales\us_english %SYBASE_HOME%\locales\english

copy -r %SYBASE_HOME%\locales\message\us_english
%SYBASE_HOME%\locales\message\english
```

Otherwise you may get the following Sybase error:

```
Using default locale name: "default"
Cannot access C:\dev\sybase15\locales\message\english\common.loc file
Cannot access C:\dev\sybase15\locales\message\english\cslib.loc file
```