



Apelon, Inc.
Suite 202, 100 Danbury Road
Ridgefield, CT 06877

Phone: (203) 431-2530
Fax: (203) 431-2523
www.apelon.com

Apelon Distributed Terminology System (DTS)

DTS Quick Start Guide

Table of Contents

Install Apelon DTS	3
Install Apelon DTS.....	3
Component and Database/Driver Selection	3
Create DTS Schema.....	6
Overview	6
Create Oracle Database User and Tablespace.....	6
Create SQL Server Database File.....	7
Create DTS Knowledgebase Schema.....	8
Create DTS User.....	8
Create User/Assign Namespace Permissions With User Manager Application.....	8
Change User Password (User Manager GUI Application)	8
Delete DTS User (User Manager GUI Application).....	8
Populate DTS Knowledgebase Tables.....	8
Perform “Full” Import Using the DTS Subscription Import Wizard	8
Perform “Full” Import Using kbcontent-import.bat	8
Populate New Tables Through Data Migration.....	8
Run Knowledgebase Admin.....	8
Configure Apelon DTS Server.....	8
Start Servers and DTS Browser	8
Start Apelon DTS Server.....	8
Start Tomcat Server	8
Start Apelon DTS Browser.....	8
Uninstall Previous DTS Versions (Optional)	8

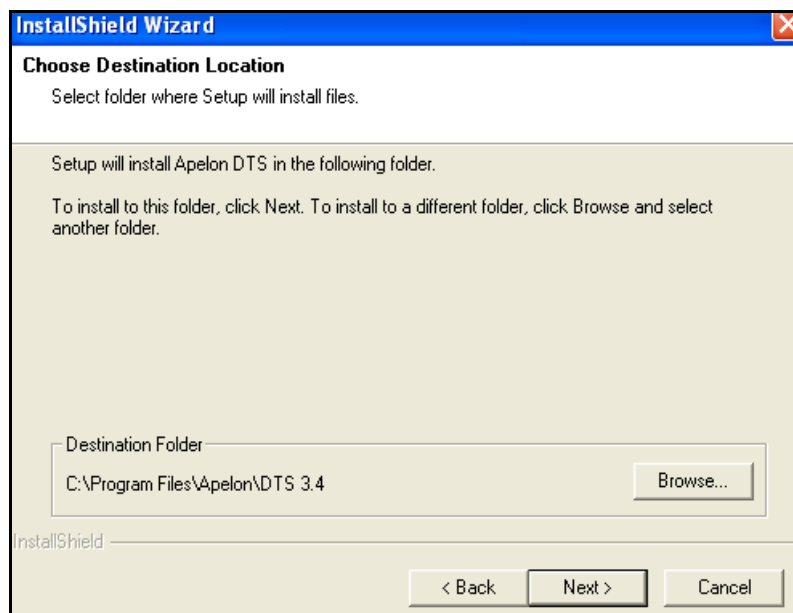
Install Apelon DTS

Refer to the *DTS Installation Guide* for more information on installing DTS.

Install Apelon DTS

Follow this procedure to install the current version of DTS. Note that you can retain previous versions of DTS on the same machine (i.e., there is no requirement to uninstall earlier DTS versions). Refer to the [Uninstall Previous DTS Versions \(Optional\)](#) discussion later in the guide for uninstall procedures.

1. From your custom Apelon Web page, **download** the current version of Apelon DTS to a local drive on your *Windows* PC, then **save** the current DTS version.
2. Click the **Apelon DTS** link to run the **InstallShield** wizard, which guides you through the installation process.



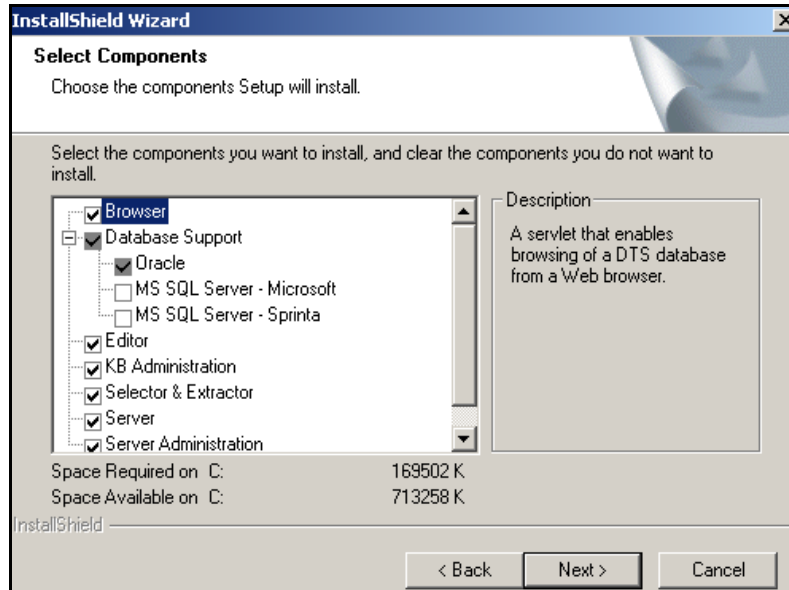
Note that default installation path. It will be referred to as *DTSInstall* throughout this document. Earlier versions of DTS, installed in other folders, may remain on the same machine.

3. Click **Next** to install Apelon DTS in the default destination folder, or click **Browse** to select another folder.

Component and Database/Driver Selection

1. As part of the installation, you are asked to select which of the DTS components (**Browser, Editor, KB Administration, Selector & Extractor, Server, Server Administration**) you want installed. The default is to install all of the DTS components; click the corresponding checkbox to deselect undesired components.

You can install the DTS components in various configurations, on the same machine, or each on separate machines (as long as all the machines containing the DTS components are connected by a network). The default is to install all six components on the machine on which the installer is run.



For **Browser** and **Editor** details, refer to the *DTS Browser Users Guide* and *DTS Editor Users Guide*, respectively. Refer to the *Knowledgebase Administrators User Guide* for details on **KB Administrator** and **Selector & Extractor**. You must install the **Server Administration** component to use the DTS User Manager application. Refer to the *DTS Server Operations Guide* for User Manager details.

Under **Database Support**, you must specify the database on which you maintain your DTS Knowledgebase, and the driver through which the DTS Server and the DTS Editor will access the data. The appropriate DTS configuration files are updated to reflect this selection.

- **Oracle** (the default) applies if you have installed either **Oracle 8i** or **Oracle 9i** to maintain your knowledgebase.
- Click **MS SQL Server – Microsoft** if you will maintain the knowledgebase in a *Microsoft SQL 2000 Server* database, and use the *Microsoft JDBC* driver.
 - An **SQL Server 2000 Type 4 JDBC** driver is required.
 - The default *Microsoft* driver is installed automatically.

- Click **MS SQL Server - Sprinta** if you will maintain the knowledgebase in a *Microsoft SQL 2000 Server* database, and use the I-Net Sprinta JDBC driver.
 - Note: For **Microsoft SQL 2000 Server** connections, Apelon recommends that you install the **iSprinta Enterprise™** driver (not included).
 - The **iSprinta Enterprise** driver provides the multiple connections required to run and use DTS.

To run classification of Ontylog Extension namespaces, you must install the **Modular Classifier** for each Ontylog subscription namespace that is linked to an Extension namespace. You must select the **Modular Classifier Support** component during installation to install the **Modular Classifier Deployment Wizard**. Refer to the *Modular Classifier Installation and Setup* discussions in the *Ontylog Extension Namespaces and Extension Namespace Classification in DTS* document for procedures.

2. Click **Next** to begin installing the selected components in the location you specified. When the *Install Complete* window displays, click **Finish**.

Create DTS Schema

Overview

Regardless of where you will obtain the data to populate the DTS tables (either by migrating data from existing databases, or performing a complete import of new subscription data) you first must run the **Knowledgebase Create** utility. This utility generates the DTS schema, creating the tables (for Oracle users) or the database (for SQL Server users) that will be populated with your imported subscription data or migrated data.

Prior to running Knowledgebase Create you must create a user and tablespace (for Oracle) or a user and database (for Microsoft SQL Server). Procedures for running these utilities are provided in the discussions that follow. The procedures for creating the DTS Knowledgebase then are provided for both Oracle and SQL 2000 users.

Create Oracle Database User and Tablespace

You must define a user and create a tablespace before the DTS 3.5 tables can be created. Navigate to the *DTSInstall*\scripts\Oracle directory (where *DTSInstall* represents the current DTS version), and run the **userCreate.bat** file. You are prompted for the following information; note that the defaults are from the previous user and tablespace that were defined.

```
This utility creates a DTS Oracle user.
Please Type the user information OR Press Enter to use the default in brackets.
Enter DTS oracle user (wayne):
Enter password for DTS oracle user (wayne):
Enter tablespace name (wayne):
Enter tablespace datafile directory (D:\Oracle\oradata\HIDE):
Enter tablespace datafile name (wayne.dbf):
Enter datafile initial size (25M):
Enter datafile autoextend size (100M):
Enter datafile maximum size (10000M): _
```

Enter DTS oracle user: Specify the name of the new Oracle user who will have the DTS schema

Enter password for DTS oracle user: Enter the password for the new Oracle user that you specified

Enter tablespace name: Enter the name of the tablespace which will hold the DTS schema for the specified Oracle user

Enter tablespace datafile directory: Specify the directory of the datafile in which the tablespace will be created

Enter tablespace datafile name: Specify the name of the datafile which the specified tablespace will use

Enter datafile initial size: Specify the size of the datafile, in bytes (use K or M to specify the size in kilobytes or megabytes, respectively)

Enter datafile autoextend size: Specify the size, in bytes, of the next increment of disk space to be allocated automatically to the datafile (use K or M to specify this size in kilobytes or megabytes, respectively)

Enter datafile maximum size: Specify the maximum disk space allowed, in bytes, for automatic extension of the datafile (use K or M to specify this size in kilobytes or megabytes, respectively)

At this point press **Enter**. You are prompted for the **password**:

```
Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.  
Enter password:
```

The new Oracle DTS user and tablespace are created.

Create SQL Server Database File

1. If Microsoft SQL Server is to be used, you must make the following edits in the **drivers.xml** file (*DTSInstall\bin*):
 - Set the **driver_count** property value to **2**.
 - Set the **display_name** value to **2** for the Microsoft SQL Server driver or the I-Net Sprinta driver.

```
<?xml version="1.0" encoding="UTF-8"?><!DOCTYPE apelprops SYSTEM  
"http://apelon.com/dtd/properties/apelprops.dtd">  
<apelprops>  
  
  <property name="driver_count" value="1" />  
  
  <!-- Oracle driver -->  
  
  <property name="display_name1" value="Oracle - Thin Client Driver" />  
  <property name="db_type1" value="oracle" />  
  <property name="jdbc_driver1" value="oracle.jdbc.driver.OracleDriver" />  
  <property name="url_template1" value="jdbc:oracle:thin:@[HOST]:[PORT]:[DATABASE]" />  
  
  <!-- MS SQL Server drivers -->  
  
  <!-- If you use MS SQL Server and install a driver for that database  
  (Apelon highly recommends the use of the Sprinta2000 driver),  
  change the above value of driver_count from "1" to "2", and uncomment the  
  appropriate section below -->  
  
  <!-- I-Net Sprinta -->  
  <!--  
  <property name="display_name2" value="SQL Server - Sprinta Driver" />  
  <property name="db_type2" value="sql2k" />  
  <property name="jdbc_driver2" value="com.inet.tds.TdsDriver" />  
  <property name="url_template2" value="jdbc:inetdae:[HOST]:[PORT]?database=[DATABASE]" />  
  -->  
  <!-- Microsoft SQL Driver configuration -->  
  <!--  
    <property name="display_name2" value="SQL Server - Microsoft" />  
    <property name="db_type2" value="sql2k" />  
    <property name="jdbc_driver2"  
value="com.microsoft.sqlserver.jdbc.SQLServerDriver" />  
    <property name="url_template2"  
value="jdbc:sqlserver://[HOST]:[PORT];database=[DATABASE];forwardOnlyMethod=serverC  
ursor" />  
    -->  
</apelprops>
```

Create DTS Knowledgebase Schema

After you create the Oracle tablespace, or SQL Server database file, you must create the DTS tables, or **schema**. Before you can run the Knowledgebase Create utility to create the schema, you must follow this procedure to configure the parameters by which the schema will be created, and by which a connection will be made to the DTS database.

1. Open the **kbcreate.xml** file (*DTSInstall\bin\kb\create*) where you set the values for **either** an Oracle **or** an SQL 2000 connection. Note the illustration.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE DBConfig SYSTEM "http://apelon.com/dtd/util/db/dbconfig.dtd">
<DBConfig>
  <!-- ORACLE CONNECTION -->
  <connection>
    <property name="type" value="oracle"/>
    <property name="user" value="dts"/>
    <property name="pass" value="dts"/>
    <property name="host" value="localhost"/>
    <property name="databaseName" value="ORCL"/>
    <property name="databasePort" value="1521"/>
    <property name="jdbcDriver" value="oracle.jdbc.driver.OracleDriver"/>
    <property name="url_template" value="jdbc:oracle:thin:@[HOST]:[PORT]:[DATABASE]"/>
  </connection>
  <!--
    #####
    # Database Connection (sql2k) #
    #####
  -->
  <!-- Microsoft Driver -->
  <!--
    <connection>
      <property name="direction" value="target" />
      <property name="type" value="sql2k" />
      <property name="user" value="sa" />
      <property name="pass" value="" />
      <property name="host" value="localhost" />
      <property name="databaseName" value="dts" />
      <property name="databasePort" value="1433" />
      <property name="jdbcDriver" value="com.microsoft.sqlserver.jdbc.SQLServerDriver" />
      <property name="url_template"
value="jdbc:sqlserver://[HOST]:[PORT];datasource=[DATABASE];forwardOnlyMethod=serverCursor" />
    </connection>
  -->
```



```

<!-- I-Net Sprinta Driver -->
<!--
    <connection>
        <property name="direction" value="target" />
        <property name="type" value="sql2k" />
        <property name="user" value="sa" />
        <property name="pass" value="" />
        <property name="host" value="localhost" />
        <property name="databaseName" value="dts" />
        <property name="databasePort" value="1433" />
        <property name="jdbcDriver" value="com.inet.tds.TdsDriver" />
        <property name="url_template"
value="jdbc:inetdae:[HOST]:[PORT]?database=[DATABASE]" />
    </connection>
-->

<!--
    If this knowledgebase will be primarily used as a staging database
    for creating incremental subscriptions, then provide the
    Master knowledgebase name in the property for masterKb,
    which has the latest version of vocabularies and against which
    subscription diff process will be done.

    This is needed to synchronize the global sequence with one on
    Master knowledgebase to avoid any IID collisions.

    Note:
    - Expects Master knowledgebase to exist in the same database instance
      as the staging.
    - If this value is left blank then the numerical value provided
      in kbcreate.bat switch -startSeq is used to seed the
      global sequence generator.

    Example:
    - For Oracle, its the user name (e.g dts)
    - For SQL2000, its the databaseName (e.g dts.dbo)
-->
    <property name="masterKb" value="" />
</DBConfig>

```

The connection parameters for **SQL2000 CONNECTION** are commented out as the default. To establish a **SQL2000** connection, comment out the **ORACLE CONNECTION** section. Uncomment out the **SQL2000 CONNECTION** section, then select either the **Microsoft Driver** or the **I-Net Sprinta Driver** to set connection parameters.

For the **user** and **pass** property values (highlighted) enter a valid username and password for the user you created when you executed **userCreate.bat**. When you run the Knowledgebase Create utility, a database connection will be established based on the values in this file.

2. Save the **kbcreate.xml** file.

3. Run the file **kbcreate.bat** (*DTSInstall\bin\kb\create*) to start the Knowledgebase Create utility. If this is the first time **kbcreate.bat** is being run, the utility creates all of the tables required to load data into the knowledgebase.

Note: If your intent is to subsequently publish this database, or use it to generate a subscription for local namespaces, you must “seed” the sequence generator differently on this database (publishers) and, in turn, the subscriber database. The numbers **1 – 20,000,000,000** are reserved for Apelon’s subscription content. As an example, the publisher sequence should **start** with **20000000001**; the subscriber (i.e., the imported of your local content) should **start** with **40000000001**.

Refer to *Appendix A - Publish Client’s Local Namespace* for local namespace content publishing procedures.

If **kbcreate.bat** was run previously, the Knowledgebase Create utility checks for existing tables. **Important Note:** If one or more tables are missing from an existing set, the utility automatically drops all existing tables, then recreates them all (without displaying any user prompts).

If all the tables exist, the utility prompts you to drop, or not drop, existing tables.

All the tables exist. Do you want to drop all tables (y/n)?

If you type **y**, the utility drops all existing tables, then recreates them all. Type **n** to bypass the drop of existing tables and creation of new ones, and to exit the utility.

The results of running the Knowledgebase Create utility are written to the log file (the default log file is **kbcreate.log** in *DTSInstall\bin\logs*. If a different log file name and location are specified in the log configuration file **kbcreatelog.xml** (*DTSInstall\bin\kb\create*) the results are written to that file.

If an error occurs in the knowledgebase creation process, the utility stops and updates the log file to reflect the error.

After generation of the DTS tables, you must load data into those tables using one of two methods. Using the first method, you populate the tables by performing a **full subscription import** of initial DTS content from a file that is exported to you from Apelon. This file contains data from the source vocabularies to which you subscribe.

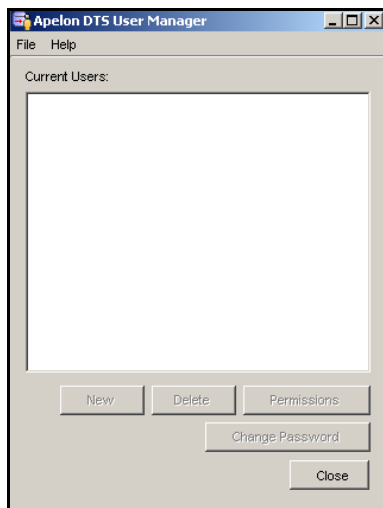
Create DTS User

If the Apelon DTS Server is to be operated in Secure Server Mode, all users must log in to use the server. Apelon DTS provides the tools to create users, and to assign and maintain each user's edit permissions within local namespaces.

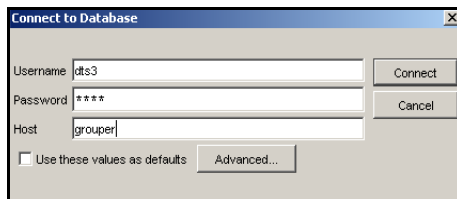
Create User/Assign Namespace Permissions With User Manager Application

A GUI-based **User Manager** tool is available from the **DTS Start** menu to perform user maintenance. Follow this procedure to use the User Manager tool to create a new DTS user, and also to assign the user edit permissions in **local namespaces**.

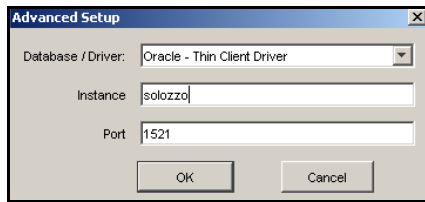
1. Select **User Manager** from the **DTS Start** menu (**Start>Programs>Apelon>DTSInstall>User Manager**). The *Apelon DTS User Manager* window displays.



2. At this point you must connect to the DTS database. Select **Connect** from the *Apelon DTS User Manager* window **File** menu. The *Connect to Database* window displays.



3. Enter your *Username*, *Password*, and *Host* name (for the machine where your database is installed). To recall these values for future sessions, click *Use these values as the defaults*.
4. If your configuration requires you to modify the default Oracle instance and port designations, or to choose SQL Server, click on **Advanced**. The *Advanced Setup* window displays.

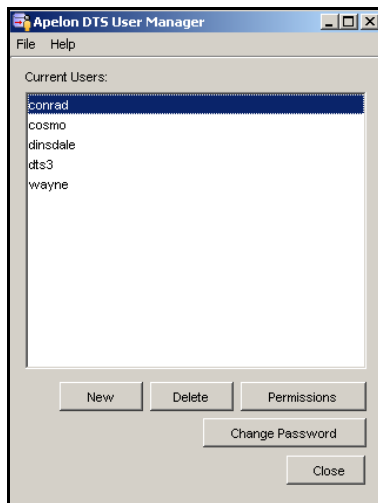


If you choose **Oracle – Thin Client Driver** in the *Database/Driver* field, you can accept the default database name in the *Database name/instance* field, and the default port number in the *Port* field, or override the defaults to connect to the desired database. Click **OK**.

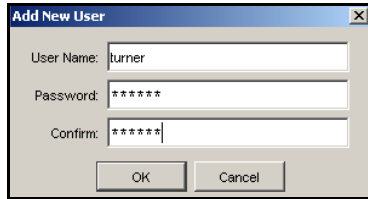
- If you indicated during DTS installation that you maintain your DTS knowledgebase in a *Microsoft SQL 2000 Server* database, and you have obtained and installed the **iSprinta Enterprise** driver, then **SQL Server – Sprinta Driver** is included as an option in the *Database/Driver* field.
- If you indicated during DTS installation that you maintain your DTS knowledgebase in a *Microsoft SQL 2000 Server* database, and selected the Microsoft driver that is provided with DTS, then **SQL Server – Microsoft** is included as an option in the *Database/Driver* field.

In the displayed *Database/Driver* field, specify your SQL Server/driver selection. In the displayed *Instance* field, specify the SQL Server named instance (if you are using a default instance, leave the field blank).

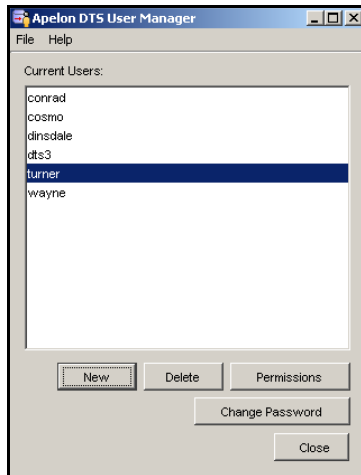
5. Click **Connect** on the *Connect to Database* window. The *Apelon DTS User Manager* window redisplay, listing current users in the **Current Users** area.



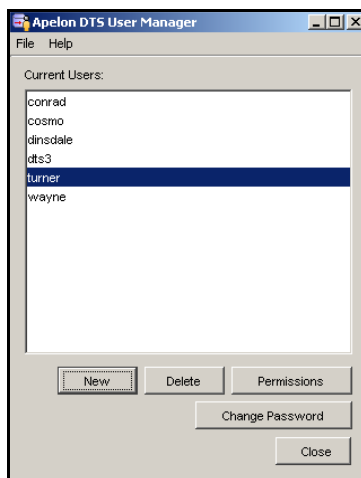
6. Click **New**. The *Add New User* window displays.



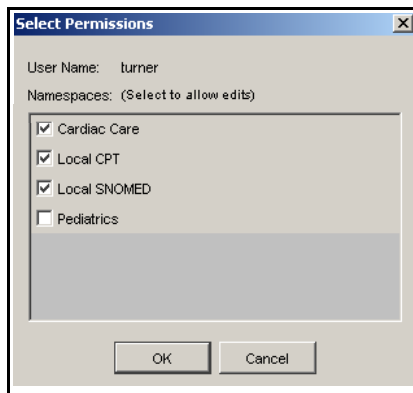
7. Specify the *User Name* and *Password* for the new user (the password displays as asterisks). Retype the new password in the *Confirm* field (the password displays again as asterisks).
8. Click **OK**. The *Apelon DTS User Manager* window displays again, listing the new user.



9. At this point you can assign local namespace edit permissions for the new user, and/or edit permissions for an existing user. Highlight a username listed under **Current Users** on the *Apelon DTS User Manager* window.



10. Click **Permissions**. The *Select Permissions* window displays. The username you selected is listed, as well as the local namespaces in your DTS Knowledgebase for which edit permissions can be assigned..

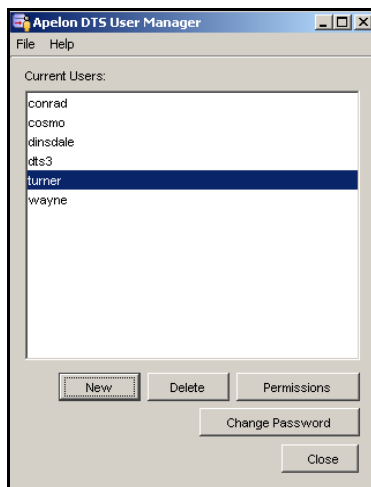


On this window you can select the **local namespace(s)** in the DTS Knowledgebase for which the user will have edit capability.

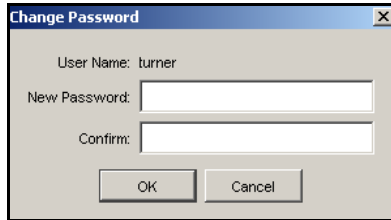
11. For the new or existing user, click each checkbox representing the local namespace for which the user will have edit permission. To remove namespace edit permission for a user, click the checkbox adjacent to the listed namespace to remove the check mark.
12. Click **OK** to create or update the user's edit permissions in the selected local namespaces. Click **Cancel** to ignore the new or updated permissions.
13. Click **Close** to exit the *Apelon DTS User Manager* window.

Change User Password (User Manager GUI Application)

1. To change an existing user's password, highlight the username on the *Apelon DTS User Manager* window.



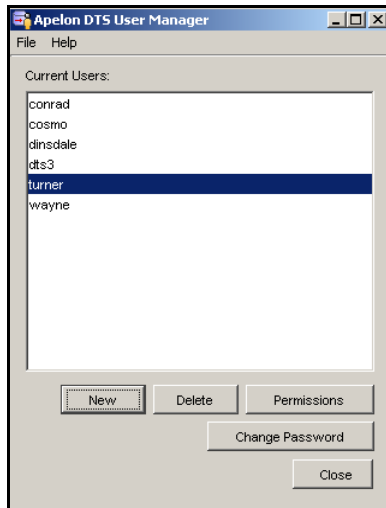
2. Click **Change Password**. The *Change Password* window displays.



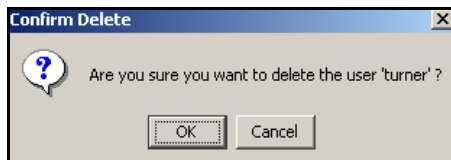
3. Enter the *New Password* for the user (the new password displays as asterisks). Retype the new password in the *Confirm* field (the new password displays again as asterisks).
4. Click **OK** to update the user password. Click **Cancel** to ignore the modification.
5. Click **Close** to exit the *Apelon DTS User Manager* window.

Delete DTS User (User Manager GUI Application)

1. To delete an existing DTS user, highlight the username on the *Apelon DTS User Manager* window.



2. Click **Delete**. The following confirmation window displays.



3. Click **Cancel** to ignore the deletion. Click **OK** to delete the username from the list of **Current Users** on the *Apelon DTS User Manager* window.
4. Click **Close** to exit the *Apelon DTS User Manager* window.

Populate DTS Knowledgebase Tables

After you create the DTS tables using the Knowledgebase Create utility, you can populate the new tables with data. If you intend to populate your DTS knowledgebase with new subscription data, refer to the [Perform “Full” Import Using the DTS Subscription Import Wizard](#) discussion. If you intend to populate the tables with migrated data from your existing knowledgebase, refer to the [Populate New Tables Through Data Migration](#) discussion.

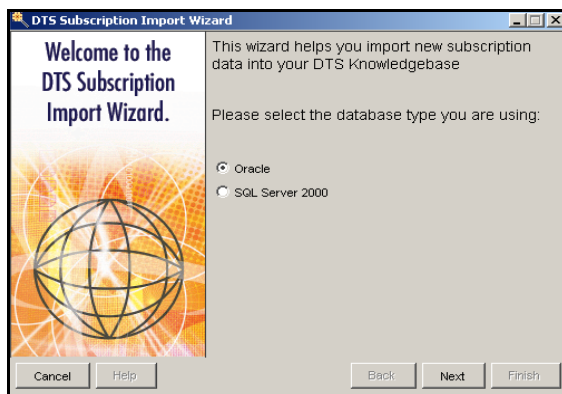
Perform “Full” Import Using the DTS Subscription Import Wizard

DTS provides a GUI-based subscription import tool called the [DTS Subscription Import Wizard](#)*. This wizard guides you through the import of initial, baseline subscription content for each of your subscription namespaces.

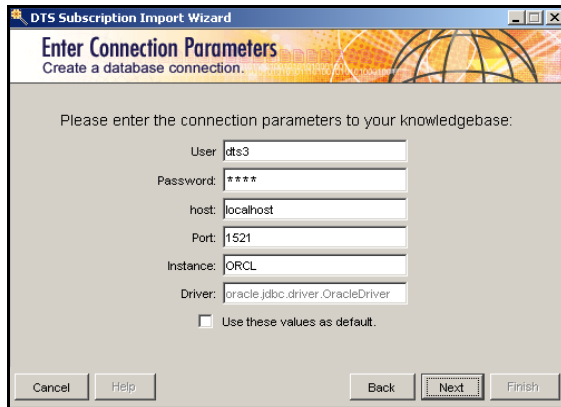
You subsequently can use the Import Wizard to [import updates](#) to your subscription namespace content. These scheduled updates are provided to you by Apelon based on your subscription agreements.

Follow this procedure to use the Import Wizard to perform a **full** import of initial subscription namespace content.

1. Download the **.zip** import subscription data file from Apelon into a directory on your machine.
2. Select **Subscription Import Wizard** from the *Windows Start* menu (**Apelon>DTSInstall>Subscription Import Wizard**). The *Welcome* window displays.



3. Ensure that the database type selected (**Oracle**, the default, or **SQL Server 2000**) reflects your DTS Knowledgebase type.
4. Click **Next**. The *Connection Parameters* window displays.

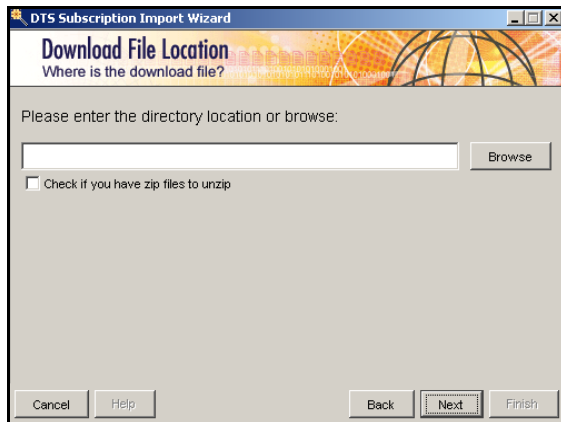


If you need to modify the database type for your DTS Knowledgebase, click **Back** to return to the *Welcome* window.

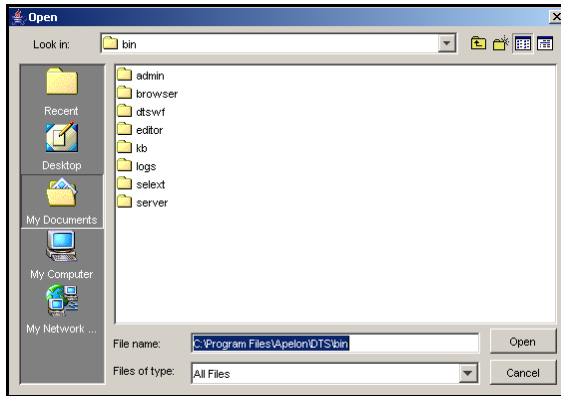
On the *Connection Parameters* window you specify the location of your DTS Knowledgebase, and the parameters for connecting to it for the purpose of importing new subscription content. The default connection parameters are from the **kbimportwizard.xml** file (*DTSInstall\bin\kb*).

You can modify the default connection parameters as necessary. Click **Use these values as default** to retain these parameters for future sessions.

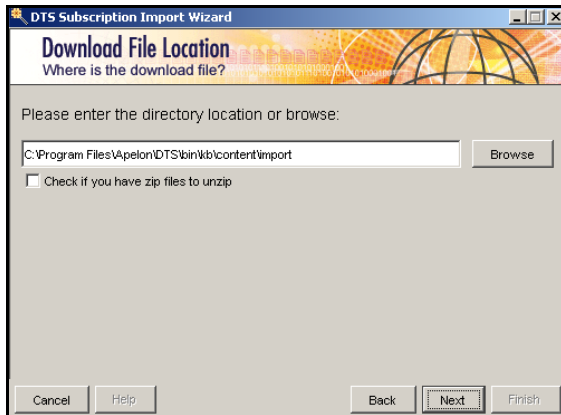
5. Click **Next**. The *Download File Location* window displays. Here you specify the directory where you want to extract (i.e., unzip) the import **.zip** file from Apelon. This directory can differ from the one where you downloaded the **.zip** file.



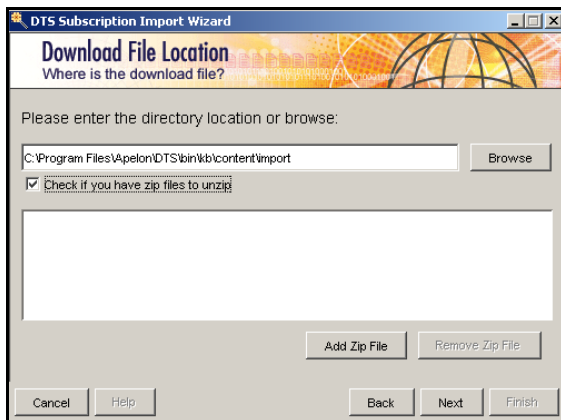
Click the **Browse** button to display the *Browse* window.



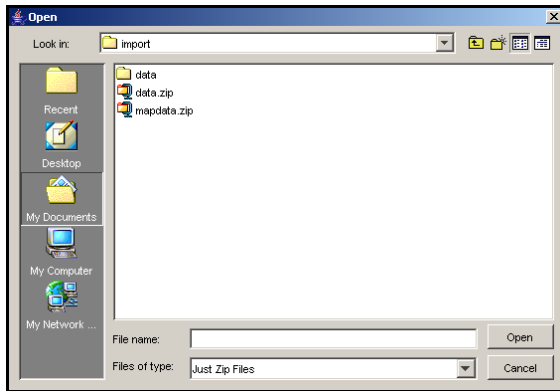
6. Browse to the directory where you want to unzip the downloaded **.zip** subscription import file (or where the downloaded file was unzipped manually) then click **Open**. The *Download File Location* window redisplay, referencing the directory path you selected.



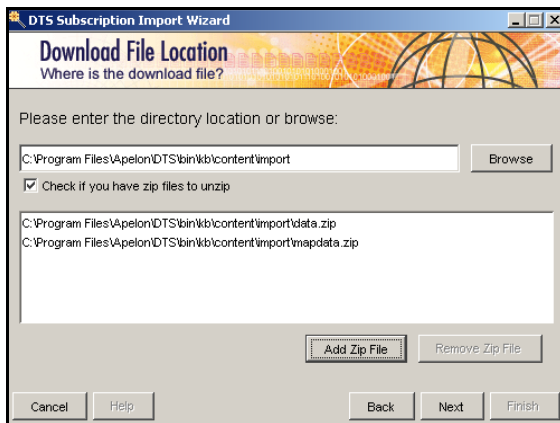
7. If you have not unzipped the **.zip** subscription import file you downloaded from Apelon, click the option box for unzipping the file (this inserts a check mark). A view area displays in the lower portion of the *Download File Location* window.



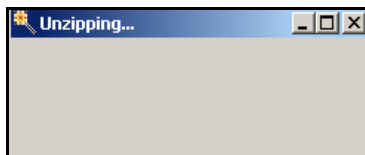
8. Click **Add Zip File**. This indicates that you want to select, and unzip, the **.zip** file that contains your subscription import information. Another *Browse* window displays.



Browse to the directory where the **.zip** subscription import file is located, click the **.zip** file name to highlight it, then click **Open**. The directory path and **.zip** file name display in the view area on the *Download File Location* window. Note that you select each **.zip** file individually; repeat this step to select multiple files.



9. Click **Next**. A window displays indicating that import files are being unzipped.

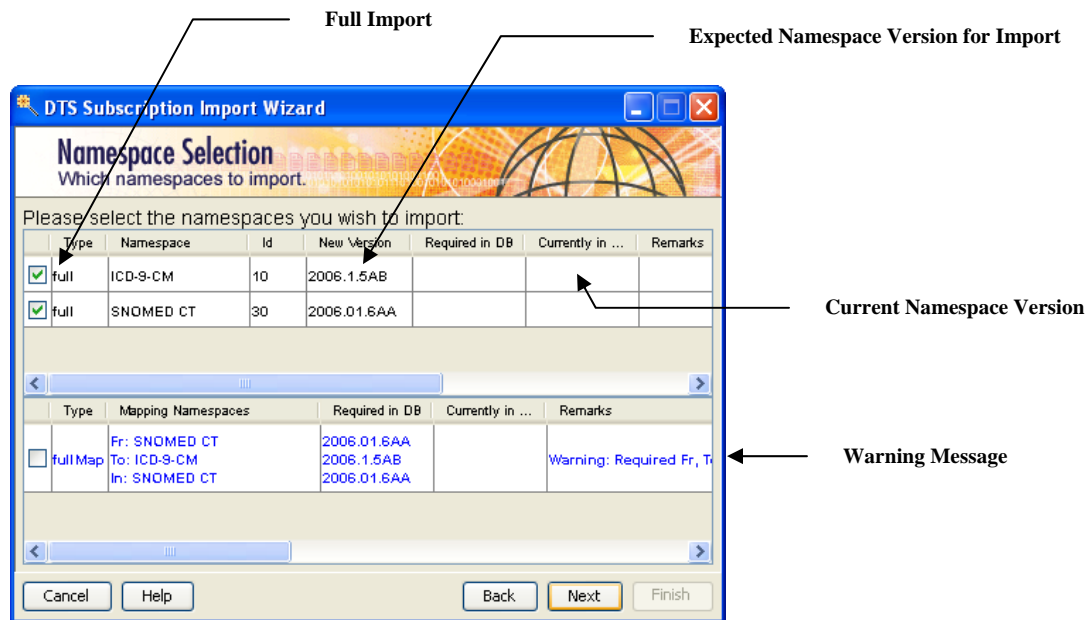


As part of the process a **data** subdirectory is created in that directory. Under the **data** directory, a separate subdirectory is created representing each namespace for which you can import content.

When the unzip process completes, the *Namespace Selection* window displays. This window lists each namespace content file (and any associated mapping file) that was included in the import **.zip** files you downloaded. The **Type** column indicates **full** for each namespace for which you are performing a full import, or **fullMap** for each full mapping import.

The latest (expected) namespace version that is required to be present in the target database is indicated under **Required in DB**. The current namespace version currently in the target database is indicated under **Currently in DB**.

In this illustration, the **Currently in DB** column does not reference a previous database (because the import is an **initial FULL** import of subscription content).



- The default is to import all listed content and mapping files from the **.zip** file. To deselect an individual file for the import, click the corresponding checkbox in the column adjacent to the **Type** column to remove the check mark.

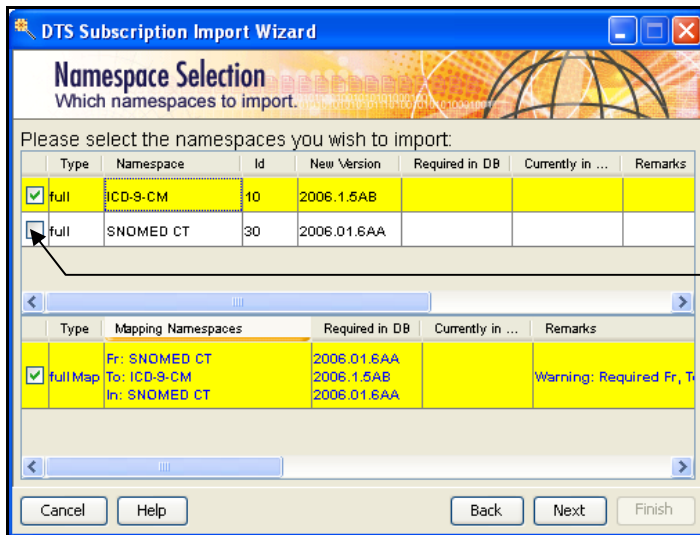
In the **Mapping Namespace** section of the window illustrated previously, the following message displays under **Remarks**, and the line is highlighted in blue text, to indicate a **Warning** scenario.

Warning: Required Fr, To or In namespaces not currently present in knowledgebase. Ensure that required namespaces are selected as part of import if not already present.

In this case, successful import of the **Mapping** namespace is contingent on the successful **Full** import of the two selected namespaces. Since **FULL** import of the two namespaces will occur first, you can check the **fullMap** box to start the import.

In each of the following illustrations, one of the FULL import namespaces is not checked for import.

Namespace Required for Mapping Import is Unchecked



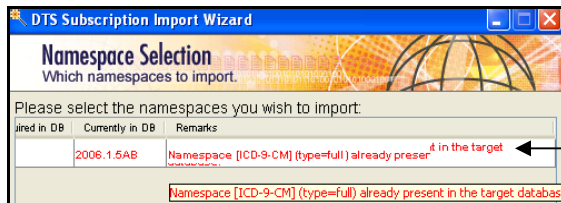
Namespace Required for Mapping Import is Unchecked

Even though you are permitted to select the **fullMap** mapping namespace for import, that import will fail unless you first select the required namespace(s) for **FULL** import.

- The subscription import content for each namespace is accompanied by a configuration file (**kbcontent-import-full.xml** for content, **kbmap-import-full-[source_nsp_id]-[target_nsp_id].xml** for mapping) that includes information regarding the version of the current target namespace (if any) into which you are importing new content. Using this information, the Import process will perform an integrity check to verify that the current (target) namespace version is compatible with the version you want to import.

The import of a specific namespace will be prevented if the process detects a version incompatibility between import namespace content and the content in the target database. The incompatibility is indicated in the *Namespace Selection* window if the values in the **Required in DB** and the **Currently in DB** fields are different. The checkbox for the namespace that is adjacent to the **Type** column is disabled, the text on that line is highlighted in red, and the **Remarks** column indicates the situation preventing the import.

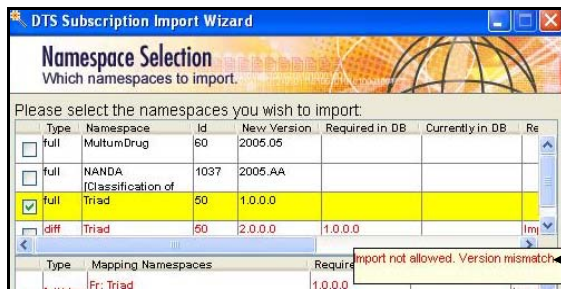
In the following illustration, the subscription content version being imported exists already in the target namespace.



Namespace Already Present in Target Database

Note that the tooltip text box also indicates the incompatibility issue.

In the next illustration, a version mismatch is indicated (i.e., the namespace version under **New Version** differs from the version under **Required in DB**). This situation can occur if successive subscription updates are missed by a client.



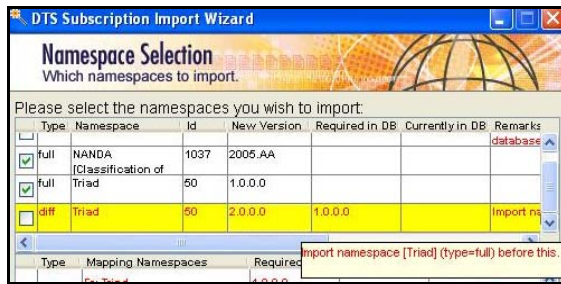
Namespace Version Mismatch

In the next illustration, an import that was interrupted earlier is being resumed (the namespace checkbox adjacent to the **Type** column is enabled in this case).



Resume Interrupted Import

The next illustration indicates that an update (i.e., **diff**) is being disabled because a **full** import of the namespace must be performed first.



Full Import Required Prior to Update

The Import process will prevent import of content for a specific namespace until you reconcile the version incompatibility referenced for that namespace.

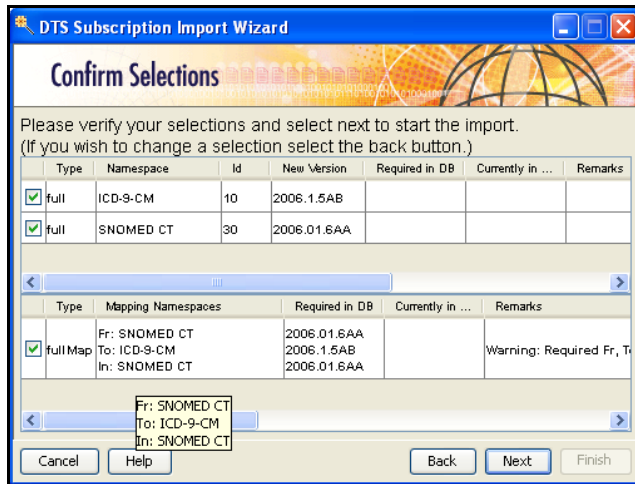
Import Content From DTS Versions Prior to 3.4

You cannot import content from DTS Version 3.3 (and older) unless there is a valid Import Integrity Verification configuration file (**kbcontent-import-full.xml** for content, **kbmap-import-full-[source_nsp_id]-[target_nsp_id].xml** for mapping) included with that older subscription content. If the configuration file is not included, the following window displays.



The Import Integrity Verification file must be generated and added to this older content before you can import that content into DTS. Refer to the *Generate Import Integrity Verification File* discussion in the *Knowledgebase Administrators Guide* for instructions on generating this configuration file, and including it with your older subscription content for import into DTS.

- When you complete namespace and mapping file import selections, click **Next** (you must make at least one namespace or mapping import selection in order to advance). The *Confirm Selections* window displays for you to verify your selections.



If you need to modify your selections, click **Back**. When the *Namespace Selection* window redisplay you can modify your selections, as needed.

13. You have the option to define the order in which namespaces will be imported. Open the **import-precedence.txt** file in *DTSInstall\bin* (the file is illustrated).

```
# Format for importing namespace
# <namespaceId>|<importType>
# Legal values for namespace importType: full or diff
# Example:
# 10|full
# 30|full
# 10|diff
#
# Format for importing mappings
# <fromNamespaceId>-<toNamespaceId>|<importType>
# Legal values for mapping importType: fullMap or diffMap
# Example:
# 30-10|fullMap
# 30-35|fullMap
# 30-10|diffMap
# 30-35|diffMap
#
# Import Options
# When set to yes, the import will not disable table constraints
# and drop indexes. By default it is set to no.
skipConstraints=no
# When set to yes, the import will run analyze tables (oracle only)
# By default it is set to yes.
runPostLoadProcesses=yes
#
# Pre-processing tasks
# Provide a file name that has tasks listed which are required to be run
# before the import process
#
# pre=
#
# Post-processing tasks
# Provide a file name that has tasks listed which are required to be run
# after the import process
#
# post=
```

List of Namespace IDs

Modify the list of source namespace ID numbers to reflect the order in which you want to import the namespaces. Note that namespaces always are imported prior to their mappings to preserve database referential integrity.

Also included in the **import-precedence.txt** file are the **-pre** and **-post** switches. You use these switches to define tasks that can be run prior to the import process, or after the import completes (e.g., addition of indexes to archive tables **prior** to the import, deletion of archive table indexes **after** import completion). You can use this feature to customize the import process with respect to performance improvements and tuning; Apelon subscriptions may include these options depending upon the size of the content updates.

Each switch must be accompanied by the task file name for the task to be run. The sample task file **pre-task-list-sample.txt** (included in *DTSInstall/bin*) defines a task for the addition of indexes prior to the import run. The sample task file is illustrated.

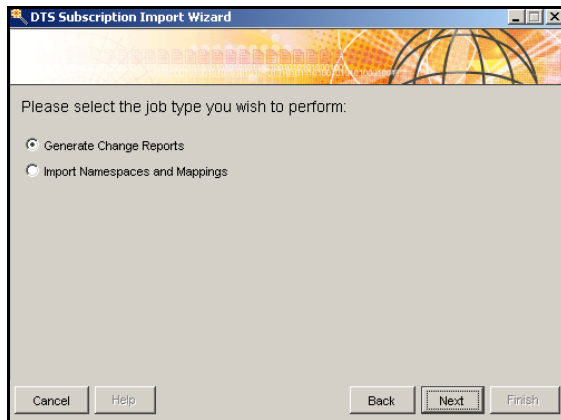
```
# Processes List Sample
#
# Format :
# executable=<executable-name>;workingDir=<working directory path (absolute)>
#
# Example:
# executable=add-my-indexes.bat;workingDir=D:\Program Files\Apelon\DTS\scripts\Oracle
```

The required entry in the **import-precedence.txt** file is illustrated.

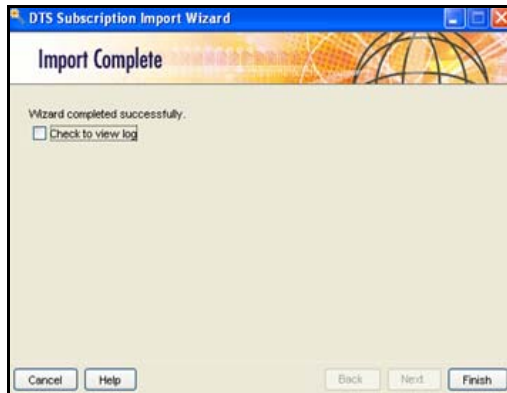
```
# Pre-processing tasks
# Provide a file name that has tasks listed which are required to be run
# before the import process ←
#
# pre=pre-task-list-sample.txt]
#
# Post-processing tasks
# Provide a file name that has tasks listed which are required to be run
# after the import process
#
# post=
```

Task File Name

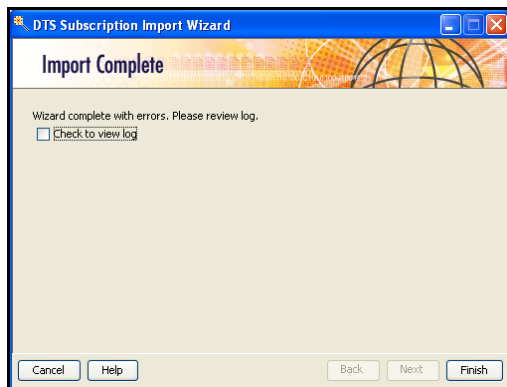
14. Click **Next**. The following window displays.



15. Click *Import Namespaces and Mappings* to begin the import process. A status window displays to indicate the progress of the import. A confirmation window displays when the import is completed.



If errors occurred during the import, click the *View Log* option to review them.



16. Click **Finish** to exit the Subscription Import Wizard.

If you cancel an import that is in progress by clicking the **Cancel** button, you can resume the import at the point at which it was stopped. Click *Import Namespaces and Mappings* to restart the import process at the point where it was interrupted.

* A modified version of the open source Java Wizard Framework was used in development of the DTS Subscription Import Wizard, which is included in this release. The Java Wizard framework and its use are covered by the GNU Lesser General Public License. A copy of the license is included in the **license.txt** file in the following installed DTS subdirectory:

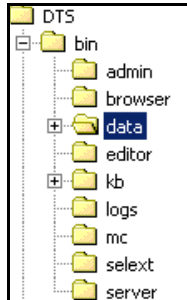
DTSInstall\samples\wizard\license.txt

You also can click the following link to view the license:

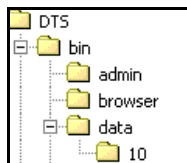
<http://www.opensource.org/licenses/lgpl-license.php>.

Perform “Full” Import Using kbcontent-import.bat

1. Unzip the **SOURCE-VERSION-FULL.zip** import subscription data file provided by Apelon into the *DTSInstall\bin* directory. A **data** subdirectory is created (if not present already) in *DTSInstall\bin*.



Under the new **data** directory, a separate subdirectory is created for each terminology source (i.e., namespace) in your knowledgebase. Note the illustration.



2. At this point you must edit the **kbcontent-import.bat** file (*DTSInstall\bin\kb\content\import*) for the import.

The subscription import content for each namespace is accompanied by a configuration file (**kbcontent-import-full.xml** for content, **kbmap-import-full-[source_nsp_id]-[target_nsp_id].xml** for mapping). Each file includes information regarding the version of the current target namespace (if any) into which you are importing new content. The configuration file for each namespace to be imported will be in *DTSInstall\bin\data*.

Using this information, the Import process will perform an integrity check to verify that the current (target) namespace version is compatible with the version you want to import. The import of a specific namespace will be prevented if the process detects a version incompatibility between import namespace content and the content in the target database.

Note the highlighted switches in the illustrated **kbcontent-import.bat** file.

```

@echo off
REM
REM Process Arguments : -p <properties-file.xml> -c <configuration full-class name>
REM
REM
REM Process arguments overriding the properties file configuration variables:
REM (i.e., connection params from a separate file overrides one in the property file):
REM Usage: -c <system configuration class name> which is mandatory
REM      -action [export, import, export_mappings, import_mappings, content_change_report, publish_namespace]
REM      export -> exporting content from DTS
REM      import -> importing content into DTS
REM      export_mappings -> exporting inter-source mappings (associations) from DTS knowledgebase
REM      import_mappings -> importing inter-source mappings (associations) into DTS knowledgebase
REM      content_change_report -> to generate reports showing concept associations affected by subscription
content update
REM      publish_namespace -> publish a local namespace
REM      -p <user property file name>
REM      XML file having the properties needed to run this process (supplied with namespace content).
REM      -skipConstraints (yes/no)
REM      If yes, the process does not disable/enable constraints and drop/add indexes while updating DTS
knowledgebase
REM      -t <target connection config file>
REM      file with the target connection params.
REM      The application loads connection params from this file, and NOT from kbcontent-import-
{full/diff}.xml
REM      -d <data extract directory>
REM      path to folder where subscription content is to be read from/written to.
REM      -runPostLoadProcesses <post load processing (yes/no) {optional}, default=yes>
REM      [e.g. analyze tables in case of Oracle]
REM      -pre <pre processes file name>
REM      -post <post processes file name>
REM
REM -- Imports a given namespace (full)

cd ..\..\

REM
REM Switch -t kb/target-connection.xml is required here as the connection parameters are read from the target-
connection.xml file.
REM
call runApp_cw 512 com.apelon.dts.db.admin.DbContentMgr -c com.apelon.dts.db.admin.config.DbContentMgrConfig -p
data/{namespace_id}/kbcontent-import-{full/diff}.xml -action import -t kb/target-connection.xml
cd kb\content\import

```

Modify -t kb/target-connection.xml Switch

Modify -p data/{namespace_id}/kbcontent-import-{full/diff}.xml Switch

Edit the **-p data/{namespace_id}/kbcontent-import-{full/diff}.xml** switch to reference the import configuration file in *DTSInstall\bin\data* that accompanied the namespace content; include a switch for each source namespace.

You can edit the **kbcontent-import.bat** file to reference a custom connection configuration file that will not be overwritten during the subscription import process. The template connection file **target-connection.xml** allows you to define a custom connection based on the **target** connection parameters for content import. To establish a target connection in the **target-connection.xml** file, the value of the **direction** property should be **target**.

Note the illustration.

```
<property name="direction" value="target" />
```

You can create multiple versions of the **target-connection.xml** file, each with its own specific name, and each reflecting a specific connection from, or to, a data source.

Modify the **-t kb/target-connection.xml** switch to reference the appropriate target-connection.xml file. Apelon recommends that you rename each modified kbcontent-import.bat batch file to reflect its purpose (e.g., **kbcontent-import-test.bat**, **kbcontent-import-production.bat**, etc.).

3. Run **kbcontent-import.bat** (*DTSInstall\bin\kb\content\import*) to import initial subscription content into your knowledgebase. Content from each namespace source referenced in the **kbcontent-import.bat** file is imported from its corresponding subscription namespace subdirectory under *DTSInstall\bin\data*.

If the Import process detects a version incompatibility that would cause the individual namespace import to fail, the import of that namespace is prevented. You must reconcile the version incompatibility for the namespace that failed the import, then attempt the import again.

Import Content From DTS Versions Prior to 3.4

You cannot import content from DTS Version 3.3 (and older) unless there is a valid Import Integrity Verification configuration file (**kbcontent-import-full.xml** for content, **kbmap-import-full-[source_nsp_id]-[target_nsp_id].xml** for mapping) included with that older subscription content.

This Import Integrity Verification file must be generated and added to this older content before you can import that content into DTS 3.4+. Refer to the *Generate Import Integrity Verification File* discussion in the *Knowledgebase Administrators Guide* for instructions on generating this file, and for including it with your older subscription content for import into DTS.

Populate New Tables Through Data Migration

If you intend to populate your new DTS tables with migrated data, you must run the **Knowledgebase Load** utility to migrate knowledgebase data from DTS 2.x or TDE data sources. The procedures referenced in this discussion represent a “**standard migration**” of TDE or DTS 2.x data into DTS 3.x.

1. Each TDE or DTS 2.x source terminology you wish to migrate to DTS 3.x must be listed in the **sources.id** file in the *DTSInstall\bin\kb* directory. A portion of the file that is included with the DTS installation is illustrated.

```
#####  
##                               ##  
## Ontylog Sources ##  
##                               ##  
#####  
  
### ICD (International Classification of Diseases, Ninth Revision,  
Clinical Modification)  
ICD-9-CM=10  
  
### CPT (Physicians' Current Procedural Terminology)  
CPT=20  
  
### SNOMEDCT  
SNOMED CT=30  
  
### SNOMED Historical Relationships (Inactives)  
SNOMED CT Inactive=35
```

If you want to migrate a local source terminology that is not listed in the **sources.id** file, you must add the source terminology with a unique ID number. The starting ID number for assignment is listed near the end of the file (e.g., **value >= 32768**).

2. To configure the connection between the source database and target database (i.e., the load destination) open the **kbload.xml** file (*DTSInstall\bin\kb\load*).

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE DBConfig SYSTEM "http://apelon.com/dtd/util/db/dbconfig.dtd">
<DBConfig>
  <!-- The authority for the namespace -->
  <property name="authority" value="Apelon3"/>
  <!-- The version label for content -->
  <property name="version" value="1.0.0.0"/>
  <!-- The version number for content (numerals only) -->
  <property name="versionNumber" value="1"/>
  <!-- The content type to be loaded ('ontylog' or 'thesaurus') -->
  <property name="conceptType" value="ontylog"/>
  <!-- Access mode for the constant ('readwrite' or 'readonly') -->
  <property name="accessMode" value="readonly"/>
  <!-- Namespace type ('local' or 'nonlocal')-->
  <property name="namespaceType" value="nonlocal"/>
  <!-- Namespace has semantic type properties ('true' or 'false') -->
  <property name="hasSemanticType" value="false"/>
  <!-- For Tuning
    <property name="numSourceRecords" value="10000" />
    <property name="numTargetRecords" value="5000000" />
  -->
  <!-- Migrating from database -->
  <connection>
    <property name="direction" value="source"/>
    <property name="type" value="oracle"/>
    <property name="user" value="cpt"/>
    <property name="pass" value="cpt"/>
    <property name="host" value="localhost"/>
    <property name="databaseName" value="ORCL"/>
    <property name="databasePort" value="1521"/>
    <property name="jdbcDriver" value="oracle.jdbc.driver.OracleDriver"/>
    <property name="url_template"
value="jdbc:oracle:thin:@[HOST]:[PORT]:[DATABASE]"/>
    <property name="blockSize" value="256"/>
  </connection>
  <!-- Migrating to database (dts) -->
  <connection>
    <property name="direction" value="target"/>
    <property name="type" value="oracle"/>
    <property name="user" value="dts"/>
    <property name="pass" value="dts"/>
    <property name="host" value="localhost"/>
    <property name="databaseName" value="ORCL"/>
    <property name="databasePort" value="1521"/>
    <property name="jdbcDriver" value="oracle.jdbc.driver.OracleDriver"/>
    <property name="url_template"
value="jdbc:oracle:thin:@[HOST]:[PORT]:[DATABASE]"/>
    <property name="blockSize" value="256"/>
  </connection>
  <!--
```

Note the following property values in the file:

- The value for the property **authority** is the authority name for this source.
- The value for the property **version** is the version name (i.e., label) for this vocabulary; the initial (full) vocabulary version for a first-time migration may be, for example, **1.0.0.0**. You can replace this value with any alpha/numeric version identifier.
- The value for the **versionNumber** property must increase for each subsequent vocabulary load (e.g., start with **1**, followed by **2**, **3**, etc.).
- The value for the property **conceptType** is **ontylog** for TDE and DTS 2.x data migrations.
- The value for the property **accessMode** is **readonly** for loading standard terminologies, and **readwrite** for local namespaces to make them editable in DTS.

Apelon suggests that you set the **accessMode** to **readonly** for the load of a TDE (i.e., ontolog) terminology, and then use the DTS Editor to create a separate (editable) local namespace to which new and modified content can be written. This approach prevents the inadvertent overwrite of any new local data (entered using the DTS Editor) if updated TDE content is migrated into the DTS knowledgebase at a later time.

- The value for the property **namespaceType** is **nonlocal** for subscription vocabularies. For **non-subscription vocabularies** and custom TDE migrations, the value is **local**.
 - The value for the property **hasSemanticType** is **true** for UMLS sources only.
3. In the database **source** parameters in the upper portion of the file, modify the source property values (e.g., **user**, **pass**, **host**, etc.) for either the DTS 2.x or TDE knowledgebase, as necessary.
 4. In the database **target** parameters in the lower portion of the file, modify the target property values (e.g., **user**, **pass**, **host**, etc.) for the DTS knowledgebase, as necessary.
 5. Save the **kbload.xml** file.

6. Run the file **kbload.bat** (*DTSInstall\bin\kb\load*) to start the Load utility. The utility opens the source and target database connections specified in the **kbload.xml** file, and migrates data from the source to the target knowledgebase.
 - The results of running the Knowledgebase Load utility are written to the log file (the level of logging is **INFO**). The default log file is **kbload.log** in *DTSInstall\bin\logs*. If a different log file name and location are specified in the log configuration file **kbloadlog.xml** (*DTSInstall\bin\kb\load*) the results are written there. If an error occurs in the load process, the utility stops and updates the log file to reflect the error.
 - Since TDE source database does not have any constraints in the table, DTS will generate errors if the migrated TDE data violates the constraints of the DTS schema. When duplication of data occurs, the **INFO** level indicates duplication of data in the table only once. In the **DEBUG** level, duplicate information is recorded in the log file.

Run Knowledgebase Admin

If you are migrating local namespace content from a DTS 2.x or TDE database to DTS 3.x using the **Knowledgebase Load** utility (see *Knowledgebase Load Procedures*) you must run the Knowledgebase Admin utility to prepare the namespace in the DTS knowledgebase for access by the DTS server and clients. If you imported **only** subscription namespace content by performing the full subscription import procedure (refer to the *Subscription Content Import Guide*) there is no need to run the Knowledgebase Admin utility.

The Knowledgebase Admin utility also allows you to create synonymous terms, and generate mappings between concepts with corresponding property values in any two namespaces. The utility creates database entries that are used in Ontylog class queries and Word Match searches (which work in conjunction with property searches). Mappings (associations) are generated based on the configuration you specify.

Note: The **Search** feature in the DTS Editor and DTS Browser includes a **Word Match** capability that works in conjunction with the property search. This feature lets you search for a concept based on property when you do not know the property's exact search string value. Refer to the *Configure Word Match Search* discussion later in this section to determine if database configurations are needed prior to performing the Knowledgebase Admin procedures.

1. In order to run the Knowledgebase Admin utility, you must establish the connection against which the utility will run in the **kbadmin.xml** file. A sample **kbadmin.xml** file is provided in *DTSInstall\bin\kb\admin* as a template for the creation of synonymous terms and associations. Note the **kbadmin.xml** file illustration. This file needs to be edited to run against **local** namespaces. The process should not be run against **subscription** namespaces (it has been run

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE DTSDbConfig SYSTEM "http://apelon.com/dtd/dts/dtsconfig.dtd">
<DTSDbConfig>
  <!--
    This is an example configuration file illustrating
    the various parameters needed to run KBAAdmin process.
    Users are required to modify it according to their needs
    and run KBAAdmin process.
    Note: The kbadmin should be run only on a local or
    non-subscription source
    (e.g., on a local namespace migrated from TDE)
  -->
  <!--
    Specify the connection paramters to connect
    to a DTS knowledgebase.
  -->
  <!-- Oracle Database Parameters -->
  <connection>
    <property name="type" value="oracle"/>
    <property name="user" value="dts"/>
    <property name="pass" value="dts"/>
    <property name="host" value="localhost"/>
    <property name="databaseName" value="ORCL"/>
    <property name="databasePort" value="1521"/>
    <property name="jdbcDriver" value="oracle.jdbc.driver.OracleDriver"/>
    <property name="url_template" value="jdbc:oracle:thin:@[HOST]:[PORT]:[DATABASE]"/>
  </connection>
  <!--
    #####
    # Database Connection (sql2k) #
    #####
  -->
  <!-- Microsoft Driver -->
  <!--
    <connection>
      <property name="direction" value="target" />
      <property name="type" value="sql2k" />
      <property name="user" value="sa" />
      <property name="pass" value="" />
      <property name="host" value="localhost" />
      <property name="databaseName" value="dts" />
      <property name="databasePort" value="1433" />
      <property name="jdbcDriver" value="com.microsoft.sqlserver.jdbc.SQLServerDriver" />
      <property name="url_template"
value="jdbc:sqlserver://[HOST]:[PORT];datasource=[DATABASE];forwardReadOnlyMethod=serverCursor" />
    </connection>
  -->
  <!-- I-Net Sprinta Driver -->
  <!--
    <connection>
      <property name="direction" value="target" />
      <property name="type" value="sql2k" />
      <property name="user" value="sa" />
      <property name="pass" value="" />
      <property name="host" value="localhost" />
      <property name="databaseName" value="dts" />
      <property name="databasePort" value="1433" />
      <property name="jdbcDriver" value="com.inet.tds.TdsDriver" />
      <property name="url_template"
value="jdbc:inetdae:[HOST]:[PORT]?database=[DATABASE]" />
    </connection>
  -->

```

This is the name of the namespace for which synonyms terms and/or associations with other namespace are to be created.

-->

```
<property name="namespace" value="ICD-9-CM"/>
```

<!-- Build synonymous term from a given property

- The first property element name attribute has the property definition name which is to be converted to a synonymous term. The "value" field attribute states whether this synonymous term is a preferred term or not ('true' for preferred, 'false' if not).
- The second property element allows user to add a property to the synonymous term created above. The value of the term property is specified in the 'value' attribute of this element.

Example below attaches a 'Term Type' property with a value 'Preferred Term' to all the synonymous terms created from the property 'Preferred Name' present in ICD-9-CM namespace.

Note:

- For a given namespace, only one type of synonymous term can be a preferred term.
- Once the property is translated to synonymous term corresponding property type definition and its instances are deleted from the namespace.

-->

<!-- Example to create synonymous term for ICD-9-CM -->

<!--

```
<synonym nameSpace="ICD-9-CM" >
  <property name="Preferred Name" value="true" />
  <property name="Term Type" value="Preferred Term" />
</synonym>
<synonym nameSpace="ICD-9-CM">
  <property name="Entry Term" value="false" />
  <property name="Term Type" value="Entry Term" />
</synonym>
```

-->

<!-- Example to create synonymous term for CPT -->

<!--

```
<synonym nameSpace="CPT" >
  <property name="Preferred Name" value="true" />
  <property name="Term Type" value="Preferred Name" />
</synonym>
<synonym nameSpace="CPT" >
  <property name="Synonym" value="false" />
  <property name="Term Type" value="Synonym" />
</synonym>
```

-->

<!-- Build associations (or mappings) between two concepts
having the same property values in two different namespaces.

- Mapping element attribute 'sourceNameSpace' specifies the source namespace name.
- Mapping element attribute 'targetNameSpace' specifies the target namespace name.
- Property element 'SOURCE_PROPERTY_NAME' is the property definition used to create the source end of association.
- Property element 'SOURCE_PROPERTY_NAME' is the property definition used to create the source end of association.

- Property element 'DELETE_SOURCE_PROPERTY_POST_MAP' deletes instances of any property in source namespace which are used to create associations if the value is set to 'true' or if the value is set to 'false' does not delete them.

- Property element 'DELETE_TARGET_PROPERTY_POST_MAP' deletes instances of any property in target namespace which are used to create associations if the value is set to 'true' or if the value is set to 'false' does not delete them.

Example below shows that (source) concepts with property 'SNOMED_ICD9_MAP' in SNOMED-CT and (target) concepts with property 'Code in Source' in ICD-9-CM will be mapped (associations if they have the same property value (codes in this case). After associations are created, delete all property instances for 'SNOMED_ICD9_MAP' property definitions in the SNOMED CT namespace and do not delete all property instances for 'Code in Source' property definitions in the 'ICD-9-CM' namespace.

Note:

- Both source and target namespace is required to be present before creating the associations.
- Both properties specified in the 'SOURCE_PROPERTY_NAME' and 'TARGET_PROPERTY_NAME' must be present to successfully create associations.

-->

<!--

- Property element 'ASSOCIATION_NAMESPACE_NAME' is the the name of the namespace for which the new association is to be created.

- Property element 'NEW_ASSOCIATION_NAME' is the name of the new association

- Property element 'NEW_ASSOCIATION_CODE' is the code of the new association

- Property element 'NEW_ASSOCIATION_ID' is the id of the new association

- Property element 'NEW_ASSOCIATION_CONNECTS' is the "connects" portion of the new association ("C" for association between two concepts)

- Property element 'NEW_ASSOCIATION_PURPOSE' is the "purpose" portion of the new association ("M" for mappings)

- Property element 'NEW_ASSOCIATION_INVERSE_NAME' is the inverse name of the new association (default value "NOT DISPLAYABLE")

- Property element 'REUSE_ASSOCIATION' is the boolean value to decide on whether to reuse the association if the association exists in the database.

-->

<!--

```
<mapping sourceNameSpace="SNOMED CT" targetNameSpace="ICD-9-CM" >
  <property name="SOURCE_PROPERTY_NAME" value="SNOMED_ICD9_MAP" />
  <property name="TARGET_PROPERTY_NAME" value="Code in Source" />
  <property name="DELETE_SOURCE_PROPERTY_POST_MAP" value="false" />
  <property name="DELETE_TARGET_PROPERTY_POST_MAP" value="false" />
  <property name="ASSOCIATION_NAMESPACE_NAME" value="SNOMED CT" />
  <property name="NEW_ASSOCIATION_NAME" value="SNOMED CT TO ICD9 MAPPING"/>
  <property name="NEW_ASSOCIATION_CODE" value="C100" />
  <property name="NEW_ASSOCIATION_ID" value="100" />
  <property name="NEW_ASSOCIATION_CONNECTS" value="C" />
  <property name="NEW_ASSOCIATION_PURPOSE" value="M" />
  <property name="NEW_ASSOCIATION_INVERSE_NAME" value="NOT DISPLAYABLE" />
  <property name="REUSE_ASSOCIATION" value="false" />

```

</mapping>

-->

</DTSDbConfig>

2. Modify the connection property values (e.g., **user**, **pass**, **host**, etc.) as necessary.

By default, the **Synonymous Term** block in the `kadmin.xml` file is commented out. To convert properties into synonymous terms, uncomment out that block, then modify the elements appropriately. Note that after converting to synonymous terms, the property definition is deleted from the knowledgebase.

The **Mappings** block in the `kadmin.xml` file also is commented out. To map similar property values, uncomment out that block, then modify the elements appropriately.

3. Save the **kadmin.xml** file.
4. Run the file **kadmin.bat** (*DTSInstall\bin\kb\admin*) to start the Knowledgebase Admin utility. The utility checks if the specified synonym names, namespaces, and mapping names and namespaces exist. If they do not, Knowledgebase Admin logs the error and stops.
5. Refer to the comments in the **kadmin.xml** file for additional information regarding the **synonymous terms** and **mappings** procedures.

Configure Apelon DTS Server

Access to the Apelon DTS Server is based on the configurations you define in the **apelonserverprops.xml** file (*DTSInstall\bin\server*). Within this file are the properties and configurable values for accessing a database using Apelon DTS. If any of your connection parameters differ from the values established in this file, you must edit the appropriate properties before starting the Apelon DTS server.

If you are using the Apelon DTS Server in secure mode, you can establish **Authentication** to require that users log in (with a password) to a secure socket connection. Set the **authentication** property value to **true** to require login; set to **false** (the default) if the user is not required to log into the server. Both client and server should set the property to **true** to enable a secure connection.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE apelonserverconfig SYSTEM "http://apelon.com/dtd/properties/apelonserverconfig.dtd">
<!-- ApelProps-->
<apelonserverconfig>
  <property name="port" value="6666"/>
  <property name="log" value="server/apelonserverlog.xml"/>
  <property name="validate_parser" value="false"/>
  <property name="authentication" value="false"/>
  <property name="user_admin" value="com.apelon.dts.server.DTSUsers"/>
  <connection default="true">
    <property name="type" value="oracle"/>
    <property name="jdbcDriver" value="oracle.jdbc.driver.OracleDriver"/>
    <property name="url_template" value="jdbc:oracle:thin:@[HOST]:[PORT]:[DATABASE]"/>
    <property name="user" value="dts"/>
    <property name="pass" value="dts"/>
    <property name="host" value="localhost"/>
    <property name="database_name" value="ORCL"/>
    <property name="database_port" value="1521"/>
  </connection>
  <!-- Example SQL Server connection (I-Net Sprinta driver) -->
  <!--
  <connection>
    <property name="type" value="sql2k" />
    <property name="jdbcDriver" value="com.inet.tds.TdsDriver" />
    <property name="url_template" value="jdbc:inetdae:[HOST]:[PORT]?database=[DATABASE]" />
    <property name="user" value="sa" />
    <property name="pass" value="" />
    <property name="host" value="localhost" />
    <property name="database_name" value="dts" />
    <property name="database_port" value="1433" />
  </connection>
  -->
  <!-- Example SQL Server connection (Microsoft driver) -->
  <!--
  <connection>
    <property name="type" value="sql2k" />
    <property name="jdbcDriver" value="com.microsoft.sqlserver.jdbc.SQLServerDriver" />
    <property name="url_template"
value="jdbc:sqlserver://[HOST]:[PORT];datasource=[DATABASE];forwardReadOnlyMethod=serverCursor" />
    <property name="user" value="sa" />
    <property name="pass" value="" />
    <property name="host" value="localhost" />
    <property name="database_name" value="dts" />
    <property name="database_port" value="1433" />
  </connection>
  </apelonserverconfig>
```

Start Servers and DTS Browser

Start Apelon DTS Server

When you start the Apelon DTS Server (**Start>Programs>Apelon>DTSInstall>Start Apelon DTS Server**) the window that displays includes the following to indicate that the server started successfully.

```
Starting Apelon Server on 192.168.1.35:6666
Initializing socketServer
The validation of XML parser is on
RSS: Listening on port: 6666
Initializing adminServer
Starting socketServer
```

Start Tomcat Server

To run the Apelon DTS Browser, you first must start the Tomcat Server on the machine where the DTS Browser component is installed. When you start the Tomcat Server (**Start>Programs>Apelon>DTSInstall>Start Tomcat**) this window displays to indicate the server started successfully.

```
Start Tomcat
Using CATALINA_BASE:  ..
Using CATALINA_HOME:  ..
Using CATALINA_TMPDIR:  ..\temp
Using JAVA_HOME:  ..\..\java\jdk
Starting service Tomcat-Standalone
Apache Tomcat/4.1.27
```

NOTE: If you access the DTS Browser across the Internet through a “firewall,” you may need to change the Tomcat connection port. If this is the case, open the **server.xml** file (**DTSInstall\tomcat\conf**) and change the default connection port from **8081** to a port that is open through the firewall (at some installations, port **80** may be the only one available). Note the highlighted port designation in the following illustration.

```
<Connector className="org.apache.catalina.connector.http.HttpConnector"
    port="8081"
    minProcessors="5"
    maxProcessors="75"
    enableLookups="true"
    redirectPort="8443"
    acceptCount="10"
    debug="0"
    connectionTimeout="60000"/>
```

Start Apelon DTS Browser

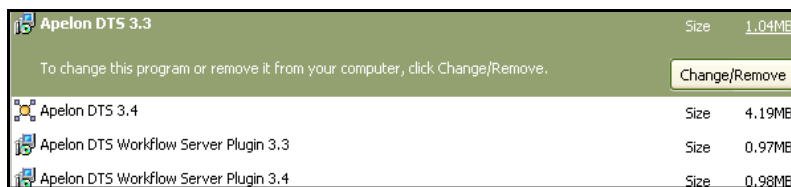
When you start the Apelon DTS Browser (**Start>Programs>Apelon>DTSInstall>Apelon DTS Browser**) a Browser window will open and you will be prompted to enter a host and port. If you are hitting the local machine, just enter **localhost / 6666**.

If you receive a “page cannot be displayed error” because you changed the default connection port from **8081** in the **server.xml** file, you must change the address (URL) to reflect the modified port. For example, change <http://localhost:8081/dtstreebrowser/index.jsp> to <http://localhost:80/dtstreebrowser/index.jsp>. You also can edit the properties on the Apelon DTS Browser link off the **Start** menu so that the proper connection will be used from that point forward.

Uninstall Previous DTS Versions (Optional)

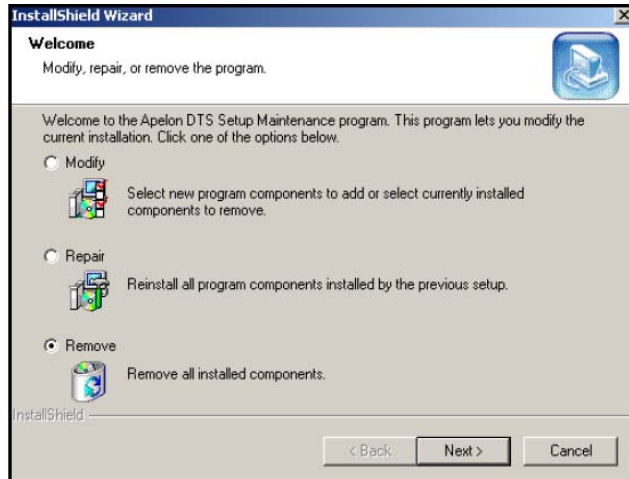
There is no need to uninstall any previous version(s) of DTS prior to installing a current version. However, if you find it necessary to delete one or more DTS versions (e.g., DTS 3.3) from your machine for maintenance purposes, use the **Add/Remove Programs** tool included with Microsoft *Windows*™ to uninstall the previous version(s).

1. Navigate to the **Start>Settings>Control** panel.
2. Select **Add/Remove Programs**.
3. Scroll down the list of programs and locate the previous DTS version you want to uninstall (e.g., **DTS 3.3**) then click to select it.



You can **individually** select any, or all, previous DTS versions to be removed.

4. Click **Change/Remove** to uninstall the selected DTS version(s). The following window displays.



5. Click **Remove**, then click **Next** to remove all components of the previous DTS version.

Note: The Add/Remove Programs tool will **not** remove any release patches you have installed for previous DTS versions. After you run the Add/Remove Programs tool to remove previous DTS versions, you must remove these patches by manually deleting the **patches** directory (*DTSInstall\lib\patches*).

[Back to Top](#)