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 3.5.1 Installation Guide
(for Linux)

Table of Contents

Introduction

The *Apelon DTS Installation Guide for Linux* includes installation procedures and requirements for the Apelon Distributed Terminology System (DTS). This guide was prepared to help new and experienced administrators successfully perform all operations necessary to install the Apelon DTS product.

Refer to the *Knowledgebase Administrators Guide* for procedures on creating the required DTS schema, and for populating the DTS tables with vocabulary subscription data provided by Apelon, data migrated from your current DTS database, or data from a TDE namespace. For instructions on how to configure and run the DTS Server, refer to the *DTS Server Operations Guide*.

Matching capabilities that are available during knowledgebase searches are described in the *DTS Matching – Quick Overview*. Preparation of the knowledgebase to support these matching capabilities is detailed in the *Knowledgebase Administrators Guide*.

Apelon DTS Installation (Linux)

This section details the procedures necessary to install and uninstall Apelon DTS on Linux. These procedures assume that the prerequisite software (listed below) has been installed, and is capable of supporting these applications.

Installation Requirements

The following are minimum system and software requirements to install and run DTS.

Minimum System Requirements (Server)

- 2GHz or above

- 2 Gigabyte RAM

- 30/40 GB free space (depending on content size)

Minimum Software Requirements (Server)

- Redhat Enterprise Linux 4 ES, Redhat Enterprise Linux 5 ES (for DTS Server only)

- Internet Explorer **7.0 or 8.0** , or Firefox 3.x

- One of the following:

    o Oracle 10g Standard or Enterprise Edition

    o Oracle 11g Standard or Enterprise Edition

    o InterSystems Cache 2010

    o IBM DB2 9.7 Workgroup or Enterprise Edition

- DTS 3.5.1 includes Tomcat 5.5

Install Apelon DTS

Follow this procedure to install DTS. The installation process creates the DTS directories on your machine and populates them with the files required to run DTS.

Note: 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 this section for uninstall procedures.

1. The DTS Linux installation is contained in a tar file called **dts_vvv.xxx-ddd.tar.gz** (where vvv represents the version number, **xxx** represents the build number and ddd represents target database – oracle, cache or db2) that you can download from Apelon DTS download page - http://apelon-dts.sourceforge.net/download.html . There is separate distribution for Oracle, Intersystem Cache and DB2. Create the target DTS directory to which you will download and uncompress this tar file.

   Note that the DTS installation directory name should **not** contain any spaces). Tomcat will fail to start if it finds spaces in the directory path.

2. Download the current version of Apelon DTS to the target DTS directory you created.

3. Untar (i.e., uncompress) the tar file from Apelon with the following command.

   <div align="center">

   tar zxvf dts_vvv.xxx-ddd.tar.gz

   </div>

   If you are installing DTS along with the DTS Workflow Server, untar that alternate file with the following command.

   <div align="center">

   tar zxvf dtswf_vvv.xxx.tar.gz

   </div>

4. Execute **sh makeScriptsExecutable.sh** from the **bin** directory. This applies execute permissions to the scripts, allowing you to run them without having to use the **sh** command.

After you uncompress the tar file, the DTS installation directory should contain the following directories:

```
[appadmin@linux DTS]$ ls
bin
conf
docs
icons
javadoc
lib
samples
scripts
tomcat
```

The **dtswf** directory also is created for DTS Workflow Linux installations.

- You must obtain and install Version 1.6 of the Sun **Java Development Kit** (JDK) or **Java Runtime Environment** (JRE) for Linux

    o Edit the login script **.bash_profile** to export environment variables **JRE_HOME** and **PATH**

    ```
    JRE_HOME=/usr/java/jre1.6.0_18
    PATH=$JRE_HOME/bin:$PATH

    export JRE_HOME
    export PATH
    ```

    o **JRE_HOME** is the directory where you installed your JDK or JRE

Install DTS as a Linux Service (Daemon)

1. Login as a superuser (root).

2. Edit the root user's **.bash_profile** file to export the environment variable **JRE_HOME**.

3. Change the directory to the one in which you installed DTS.

4. Change to the **bin** subdirectory.

5. Execute **installsrvc.sh.** You are prompted to indicate if you want to start the DTS service immediately. Enter **Yes** to start DTS as a service.

    Each time you reboot the machine, DTS will be started as a service.

    If you decide later to uninstall DTS as a Linux server, execute **uninstallsrvc.sh** (you must be root user to run **uninstallsrvc.sh**).

Starting/Configuring the Apelon DTS Server

Some server configuration may be required before you to start the Apelon DTS server. Refer to the *Server Configurations* discussions in the *DTS Server Operations Guide*.

Creating and Preparing Your Knowledgebase

A set of tools included with DTS allows you to generate the required knowledgebase tables, populate the tables with desired data, then update and maintain your Knowledgebase moving forward. Procedures for performing these tasks are provided in the *Knowledgebase Administrators Guide*.

If you have existing data in a previous version of DTS (e.g., **DTS 3.2**, **DTS 3.3**) refer to the migration information included with the *DTS Release Notes*.

Appendix A - DTS Directory Structure and Descriptions

The Apelon DTS Linux installation generates the following directory structure.

```
[appadmin@linux DTS]$ ls
  bin
  conf
  docs
  icons
  javadoc
  lib
  samples
  scripts
  tomcat
```

The script files generated for Linux installations (specifically) are referenced in the following discussions. For Linux installations, shell script files exist alongside DTS batch (.bat) files in some of the directories; you should ignore these existing batch files.

bin

The **bin** directory includes five shell script files (and one non-script Linux file) that are necessary for running Apelon DTS for Linux.

- *drivers.xml* contains information on the JDBC drivers available (Oracle and/or SQL Server)
- *dts.sh* – starts, stops or indicates the status of the Apelon DTS server (must be root user to run)
- *import-precedence.txt* contains parameters that define the order in which namespaces will be imported when you run the Subscription Import tool
- *installsrvc.sh* – installs DTS as a Linux service (must be root user to run)
- *license.txt* includes a copy of the license that covers use of the Java Service Wrapper
- *makeScriptsExecutable.sh* – this is the first shell script to run, and makes all shell scripts executable
- *shell_functions* – this file contains common shell script functions used by other shell scripts
- *subscriptionImportWizard.sh* – starts the Subscription Import Wizard
- *uninstallsrvc.sh* – uninstalls DTS as a service on Linux (must be root user to run)
- *wrapper.exe* is the Java Service Wrapper executable file (Windows only)

The **bin** directory also includes a series of subdirectories, one for each major component of DTS (e.g., Browser, Server, etc.). Within these subdirectories are the property files and scripts necessary for running Apelon DTS, scripts for running tests, and the data needed to run the tests.

bin/admin

The **admin** subdirectory contains files related to DTS Server administration, and the creation and maintenance of DTS users.

- *admin/RemoteAdmin.sh* – this script allows administrators to connect to the server to check server resources, connections, and to shutdown the server from a remote site
- *UserManager.sh* – this script provides the capability to add users for authentication on a secure socket connection to the server
- *usermanager.xml* contains the DTS Server connection settings for using the GUI version of the DTS User Manager tool (the Command Line version of the tool does not reference this file)
- *usermanagerlog.xml* contains the parameters to configure logging for the GUI version of the DTS User Manager tool

bin/browser

The **browser** subdirectory contains the *runtomcat.sh* script, which starts or stops a local copy of Apache Tomcat 5.5.16 (JRE version 1.5 or higher must be installed in order to run Tomcat). The script *runtomcat.sh* takes one argument (**start** or **stop**).

bin/kb

The **kb** subdirectory consists of scripts, files, and subdirectories related to DTS Knowledgebase connections and content.
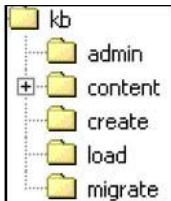
- *kbcontentlog.xml* contains the configuration parameters by which a log file is generated as a result of a knowledgebase content import or export
- *kbcontent-ora.xml* contains the data access object for Oracle that is required to run exports and imports
- *kbcontent-cache.xml* contains the data access objects for Cache that are required to run exports and imports
- *kbcontent-db2.xml* contains the data access objects for DB2 that are required to run exports and imports
- *kbcontent-tablelist.xml* lists tables that are populated with knowledgebase data
- *kbimportwizard.xml* contains the configuration parameters for the Subscription Import Wizard, which you use to import new subscription content into your knowledgebase

- *source-connection.xml* is a file, along with *target-connection.xml*, used by all the Knowledgebase utilities so the user only has to specify source and target connections in one of these files
- *sources.id* contains the names that identify namespace mappings (used in the knowledgebase load and publishing processes)
- *target-connection.xml* is a file, along with *source-connection.xml*, used by all the Knowledgebase utilities so the user only has to specify source and target connections in one of these files. By default, the parameters in these files override those in all the configuration files for each utility program.

bin/kb/admin

The **admin** subdirectory contains scripts and files related to the DTS Knowledgebase Admin utility.

- *kbadmin.sh* runs the Knowledgebase Admin utility, which creates synonymous terms, mapping between concepts with similar property values in any two namespaces
- *kbadmin.xml* contains the parameters to configure the connection against which the Knowledgebase Administration utility will run
- *kbadminlog.xml* contains the parameters to configure logging for the Knowledgebase Administration utility
- *kbadmintable.xml* lists the tables that will be populated by the Knowledgebase Admin utility

bin/kb/content/delete

The **content\delete** subdirectory contains scripts and files related to deletion of namespaces from the DTS Knowledgebase.

- *kbcontent-delete.sh* deletes a subscription namespace
- *kbcontent-delete.xml* contains the configuration parameters by which one or more namespaces can be deleted from the DTS Knowledgebase

bin/kb/content/export

The **content/export** subdirectory contains files related to the export of namespace data from the DTS Knowledgebase.

- *export-mappings.sh* exports mappings (associations) between source and target namespaces in Common Data Format (CDF) from the local client namespaces for publication to subscribers
- *export-mappings.xml* contains the configuration parameters by which **export-mappings.sh** exports inter-source mappings for publication to subscribers

- *kbcontent-export.xml* contains the export configuration parameters by which namespaces in Common Data Format (CDF) are extracted from the Master knowledgebase

bin/kb/content/import

The **content/import** subdirectory contains scripts and files related to the import of namespace data into the DTS Knowledgebase.

- *import-mappings.sh* imports mappings (associations) in Common Data Format (CDF) between source and target namespaces
- *import-mappings.xml* contains the configuration parameters by which **import-mappings.sh** imports inter-source mappings
- *kbcontent-import.sh* imports content from subscription namespaces in Common Data Format (CDF) into a client database
- *kbcontent-import.xml* contains the configuration parameters by which **kbcontent-import.sh** imports namespaces (sources) into the client database

bin/kb/content/publish

The content/publish subdirectory contains scripts and files related to the publication of namespace content from the DTS Knowledgebase.

- *kbcontent-publish.sh* allows Apelon clients to publish selected local namespace content for their own subscribers
- *kbcontent-publish.xml* contains the configuration parameters by which client namespace content is published for subscribers

bin/kb/content/report

The **content/report** subdirectory includes scripts and files related to knowledgebase content change reporting.

- *content-change-report.sh* generates the **Content Change Report**, which lists local namespace content that might be affected as a result of your subscription content import (specifically, associations linked to concepts are indicated)
- *kbcontent-report.xml* contains the database connection parameters for running the **Content Change Report**. This report lists content in a local namespace that might be affected as a result of your subscription content import (specifically, associations that are linked to concepts are indicated).

bin/kb/create

The **kb/create** subdirectory includes scripts and files related to creation of the DTS Knowledgebase.

- *kbcreate.sh* runs the Knowledgebase Create utility; if this is the first time **kbcreate.sh** is being run, the utility creates all of the tables required to load data into the knowledgebase.
- *kbcreate.xml* contains the parameters by which the schema will be created, and by which a connection will be made to the DTS database. These configurations are required before you can run the Knowledgebase Create utility.
- *kbcreatelog.xml* contains the parameters to configure logging for the Knowledgebase Create utility.

bin/kb/load

The **kb/load** subdirectory includes scripts and files related to migration of TDE data to the DTS Knowledgebase.

- *kbload.sh* runs the Knowledgebase Load utility, which loads data from a TDE namespace into the DTS tables
- *kbload.xml* contains the parameters for connecting the TDE (source) and DTS (target) databases when the Knowledgebase Load utility is run
- *kbloadlog.xml* contains the parameters to configure logging for the Knowledgebase Load utility
- *LicenseUtility.sh* Apelon users who import only subscription content from Apelon are not required to run this file. This file must be run if your company, in turn, publishes content to your own subscribers
- *namespaceStats.sh* retrieves information for namespaces in a given knowledgebase
- *namespaceStats-config.xml* contains configuration parameters for running a statistics report on one or more namespaces

bin/kb/migrate

The **kb/migrate** subdirectory includes scripts and files related to the migration of a previous version of the DTS schema to the current version.

- *kbmigrate.sh* runs the schema migration utility, which migrates a previous version of the DTS tables (e.g., 3.2) to the current DTS schema
- *kbmigrate.xml* contains the connection parameters for the database schema you intend to update when you run **kbmigrate.sh**. Note that you also can modify the **target-connection.xml** file to set the connection parameters for update of the DTS schema to the current version.
- *kbmigratelog.xml* contains the configuration parameters by which logging is performed when you run the schema migration utility (**kbmigrate.sh**)

bin/logs

The **logs** subdirectory is empty at point of installation. During the operation of Apelon DTS various log files will be generated and archived in this directory.

bin/mc

The **mc** subdirectory includes the script that runs the Modular Classification Deployment Wizard.

- *mcdeploy.sh* runs the DTS Modular Classification Deployment Wizard utility

bin/selext

The **selext** subdirectory contains scripts and files that pertain to the Selector & Extractor and MatchPack utilities.

- *filterSpec.dtd* defines the selection and extraction language used
- *filterSpec.xml* contains the parameters by which knowledgebase data will be acquired to populate generated data **silos** (a silo is a repository of related data, populated with data based on your specifications)
- *selectorlog.xml* contains the parameters to configure logging when the Selector & Extractor is run
- *selext.sh* starts the Selector & Extractor utility
- *selextconnection.xml* contains the server connection parameters for running the Selector & Extractor utility

bin/server

The **bin/server** subdirectory contains scripts and files that pertain to configuring, testing, and starting the DTS Server.

- *ApelonEncrypter.sh* runs the **Apelon Encrypter/Decrypter** tool
- *apelonserverlog.xml* contains the parameters to configure logging for the servers
- *apelonserverprops.xml* contains the properties and values to run the Apelon DTS server
- *mc.properties* contains URLs used by the Ontylog subscription namespace **Modular Classifier** using the DTS Modular Classification Deployment Wizard
- *pingtest.sh* runs ping tests based on the configuration parameters in **pingtestconfig.xml**
- *pingtestconfig.xml* contains the configuration parameters that determine which type of ping tests will be executed
- *ShutdownApelonServer.sh* shuts down any Apelon DTS Server foreground or background (i.e. running as a service) process running currently. If the process to be stopped is a background process (DTS Service), you must run this script as a root user; foreground processes can be stopped by any user.
- *StartApelonServer.sh* executes the command to start the Apelon DTS Server

- *StatusApelonServer.sh* indicates the status of Apelon DTS server

docs

The **docs** directory contains User Guides and Release Notes specific to the installed version of DTS, each in *.pdf* format. The documents are available from the *Windows* DTS **Start** menu, and can be viewed by any system that includes a copy of the Adobe Acrobat Reader.

The **ReadMe.txt** file that displays during installation also is included.

- The *dtsbrowser.pdf* is the *DTS Browser Users Guide* in Adobe Acrobat portable document format.
- The *dtsclassification.pdf* is the *Ontylog Extension Namespaces and Extension Namespace Classification in DTS* document in Adobe Acrobat portable document format.
- The *dtseditor.pdf* is the *DTS Editor Users Guide* in Adobe Acrobat portable document format.
- The *dtseditorplugins.pdf* is the *DTS Editor Plug-ins Guide* in Adobe Acrobat portable document format. .
- The *dtsmatching.pdf* is a copy of the *DTS Matching – Quick Overview Guide* in Adobe Acrobat portable document format.
- The *dtsquickstart.pdf* is a copy of the *DTS Quick Start Guide* in Adobe Acrobat portable document format.
- The *dtsreleasenotes.pdf* provides information on changes and enhancements included in the current release.
- The *dtsserver.pdf* is the *DTS Server Operations Guide* in Adobe Acrobat portable document format.
- The *dtssubscriptionimport.pdf* is the *DTS Subscription Vocabulary Content Import Guide* in Adobe Acrobat portable document format.
- The *dtssubsets.pdf* is a copy of the *DTS Subset Editor Guide* in Adobe Acrobat portable document format.
- The *dtswebdeveloper.pdf* is the *DTS Web Developers Guide* in Adobe Acrobat portable document format.
- The *kbadmin.pdf* is the *Knowledgebase Administrators Guide* in Adobe Acrobat portable document format. The guide includes instructions for creating a new DTS schema, migrating data to DTS from previous DTS versions (as well as from other databases), and using the Selector & Extractor utility.
- *dtsreadme.txt* highlights enhancements made with each release, as well as configuration and installation requirements.

javadoc

The **javadoc** directories contain the DTS client API documentation and related information. Open **Index.html** to view the HTML documentation.

lib

The **lib** directory contains the JAR files and classes necessary for running Apelon DTS.

samples

The **samples** directory contains a **dtsbrowser** subdirectory, which contains the source code for the DTS Browser. The **editorplugin** subdirectory contains the source code for the DTS Editor plug-ins. The **tutorial** subdirectory contains the source code for the DTS programming tutorial.

The **wizard** subdirectory contains a copy of the modified Java Wizard Framework code used in development of the **Subscription Wizard** and **Classification Deployment Wizard**, both of which are included in this release. You will find this source in the **wizard\src** subdirectory; in the **wizard** root directory you will find a copy of the **GNU Lesser General Public License** that covers use of this code in the **license.txt** file.

scripts

The **Oracle or DB2,** a subdirectory contains shell scripts and SQL files specific to the target database setup.

Shell scripts:

- *add-indexes.sh*
- *disable-constraints.sh*
- *drop-indexes.sh*
- *enable-constraints.sh*
- *loadContains.sh*
- *perf-analyze-tables.sh* (for Oracle installations only)
- *userCreate.sh* should be run to create the database user (note that there is a prompt for Oracle connect identifier)
    - The Connect Identifier is needed for Oracle SYSTEM user login, and usually is identical to your computer name.

SQL Files:
- *add-indexes.sql*
- *disable-constraints.sql*
- *drop-indexes.sql*

- *enable-constraints.sql*
- *perf-analyze-tables.sql* (for Oracle installations only)
- *setContains.sql*
- *userCreate.sql* - Can be run to create the database user

tomcat

Tomcat is servlet container developed by the Apache Software Foundation. It is used in the deployment of the Apelon DTS Browser included with this release. The subdirectories in the **tomcat** directory contain the files that support Tomcat. These directories and files are installed automatically when you install DTS.

Refer to the *DTS Browser Users Guide* and the *DTS Server Operations Guide* for configuration information for Tomcat.

dtswf

If along with DTS you have installed DTS Workflow, a **dtswf** subdirectory is created that contains a single script.

- *dtswfkbcreate.sh*