

**APELON**

BECAUSE TERMINOLOGY MATTERS

75 Sgt William B Terry Dr, Suite 2005, Hingham, MA 02043

+1 (203) 431-2530

[www.APELON.com](http://www.APELON.com)

[www.ApelondTS.org](http://www.ApelondTS.org)

# DTS 4: Managing DTS Editor Module Manager Facility

Table of Contents

- A. Introduction .....3**
- B. The Module Folder.....4**
- C. The InfoManager Application.....5**
  - C.1 Editing the Module Information File ..... 6
  - C.2 Setting the Module URL ..... 8

## A. Introduction

The DTS Editor **Module Manager** facility supports centralized management and distribution of DTS Editor Modules (Plug-ins). General information on available Modules is provided by a **Module Information File**. Associated **Module Kits** can be downloaded for installation. The Module Information File and Module Kits are accessed from a common Internet folder, the **Module Folder**, whose location is given by a URL called the **Module URL**.

For most DTS installations, an Apelon-hosted Module Folder gives the installation's users (clients) easy access to the latest version of standard Apelon User Modules. Some organizations, however, may wish to host their own Module Folder. Reasons to implement a private organizational folder include:

1. Eliminating client access to certain standard Modules
2. Restricting client access to specific versions of standard Modules
3. Enabling client access to other, perhaps organization-developed, Modules

The sections below present the structure of the Module Folder, and describe the tools that are available for organizations to create and manage their own folder instance.

## B. The Module Folder

The Module Folder is a network-accessible folder that contains the files necessary to support the DTS Editor's Module Manager User Module. (See the [DTS Editor User Guide](#) for a description of the Module Manager's operation.) This folder contains:

- **ModuleMgrInfo.xml**, a Module Information File that lists the available Modules
- The installation kits for all the Modules named in the Information file

The Module Information File (**ModuleMgrInfo.xml**) is an XML file that contains detailed Module data including current and historical version names, version dates, version descriptions, and Module kit (zip) and main (jar) file names. Module Information Files can be created and maintained using the DTS **InfoManager** application, available in the DTS Editor distribution kit. See the [The InfoManager](#) Application section below for instructions on the use of this application.

Module installation kits are zip files that contain all the files necessary to run a Module. Standard zip file structures are used such that if the file were to be unzipped to the DTS install directory (such as C:\Program Files\Apelon DTS 4.X) all files would be placed into the correct location in the installation. Commonly used installation folders include:

/bin/editor	configuration files
/bin/module	informational files, e.g., required .xsd and .txt files, and any .bat files supported by the Module
/docs	.pdf versions of Module documentation, e.g., User Guides
/docs/help	.htm/.html versions of Module documentation (and associated sub-folders) if accessed in the DTS Editor <b>Help</b> menu item
/lib/modules	executable .jar files

The URL for the active Module Folder (the Module URL) is maintained as a System Property in the DTS Knowledge Base. The Module URL can be set in the Knowledge Base by an option in the [The InfoManager](#) Application (see below).

### C. The InfoManager Application

The InfoManager application is provided as part of the DTS Editor installation kit to assist organizations in creating and maintaining Module Information Files. This section describes how to run the InfoManager application.

Note that the InfoManager does not directly edit files in the Module Folder. Since methods to post files in an Internet folder are inherently implementation-specific, posting is outside the scope of this document. The InfoManager creates and edits Module Information Files in a local file system, which can then be moved to the Module Folder as required.

The InfoManager is run via the InfoManager.bat file found in the /bin folder of the DTS installation. Executing this file opens the InfoManager window shown below:

The screenshot shows a window titled "File Configure" with the following fields and controls:

- Module:** A dropdown menu with a "New" button and a "Delete" button.
- Jar name:** A text input field with an "Inactive" checkbox and a "Rename" button.
- Version:** A dropdown menu with a "New" button and a "Delete" button.
- Date:** A text input field with a calendar icon and a "Rename" button.
- Supported In:** A text input field.
- Kit name:** A text input field.
- Description:** A text area with the label "Description: (html tags permitted)".
- Buttons:** "Save Module" and "Restore Module" buttons at the bottom.

The InfoManager menu bar contains two menus, the **File** menu and the **Configure** menu. The **File** menu is used to maintain Module Information Files. It has five items:

- New** Creates an empty Module Information File in the InfoManager GUI.
- Open ...** Opens a file chooser dialog for selecting an existing Module Information File for editing.
- Save** Saves the active Module Information File data in the previously opened file.

- Save As...** Opens a file chooser dialog to save the active Module Information File data in a new file.
- Exit** Closes the InfoManager application. If any data is pending, a confirmation dialog is shown.

See the *Editing the Module Information File* section below for details on editing this file's data.

The **Configure** menu contains one item: **Set Module URL ...** for setting the Module URL in the DTS Knowledgebase. See the **Setting the Module URL** section below for details on this item.

### C.1 Editing the Module Information File

The InfoManager application supports creation of new Module Information Files, but it is often easier to start with Apelon's standard file and make the necessary additions or modifications. Organizations wishing to use this option can contact Apelon Customer Support ([support@apelon.com](mailto:support@apelon.com)) for a copy of the standard file.

Once a Module Information File has been opened, the InfoManager is populated with the existing Module data:

The screenshot shows a 'File Configure' dialog box with the following fields and controls:

- Module:** A dropdown menu with 'ImportWizard' selected. Buttons: 'New', 'Delete'.
- Jar name:** A text input field containing 'ImportWizard.jar'. A checkbox for 'Inactive' is unchecked. Button: 'Rename'.
- Version:** A dropdown menu with '4.7' selected. Buttons: 'New', 'Delete'.
- Date:** A text input field containing '30-Nov-2018' with a calendar icon. Button: 'Rename'.
- Supported In:** A text input field containing '4.7'.
- Kit name:** A text input field containing 'ImportWizard-4.7.zip'.
- Description:** A text area with the label 'Description: (html tags permitted)'. The text inside reads: 'Creation of Extension Role Types in XML Namespace loads supported. Requires TQL Version 4.7|'. A cursor is visible at the end of the text.
- Buttons at the bottom: 'Save Module', 'Restore Module'.

Data presented includes the Module Name, the name of the Module's main jar file, the most recent Module Version, its Version Date, the version of DTS that supports this Module Version, the name of the Version's Kit, and a description of the Version.

To view data on another Module, select the Module's name from the **Module** dropdown. Click on the **Delete** button opposite the **Module** dropdown to delete the Module's record in the Module Information File, or click on the **Rename** button below **Delete** to rename the Module. All existing Module data is retained after a rename.

The **Jar name** is the name of the primary Module jar file: the jar that contains the class file that extends the DTSEditorModule class. The file name must end with “.jar”. Note that the **Jar name** is associated with the Module as a whole, not a specific Version. The name of this file should not be changed once a Module has been deployed.

To inactivate a Module, check the **Inactive** box. All historical data is maintained for an Inactive Module (to facilitate reporting in the Module Manager application) but the Module cannot be added or updated.

To define a new Module, click on the **New** button opposite the Module dropdown. Enter the Module name in the New Module dialog, and complete the remaining Module fields.

Operation of the Module Version elements is similar to that of the Module elements. To view data on another Version of the Module, select the Version's name from the **Version** dropdown. Click on the **Delete** button opposite the **Version** dropdown to delete the Version's record in the Module Information File, or click on the **Rename** button below **Delete** to rename the Version. All existing Version data is retained after a rename.

To define a new Version, click on the **New** button opposite the Version dropdown. Enter the Version name in the New Version dialog, and complete the remaining Version fields. Version names must be “dotted” numeric sequences, e.g., “10”, “10.5” or “10.5.1”.

The Version **Date** should be the date associated with the Version's main jar file (this is usually also the date of the kit). A variety of standard input formats are supported. Alternately, click on the calendar icon to select the date from a calendar widget.

The **Supported In** field specifies the first (earliest) version of DTS that supports this Module Version. Enter the DTS version in the same format as the Module Version. Since multiple versions of DTS may access the same Module Folder, the DTS Editor **Module Manager** uses this field to filter Module Versions so that only the most recent Version of a Module that is supported in the active DTS version is displayed. This field may be empty for existing Module Versions, but is required for new Module Versions.

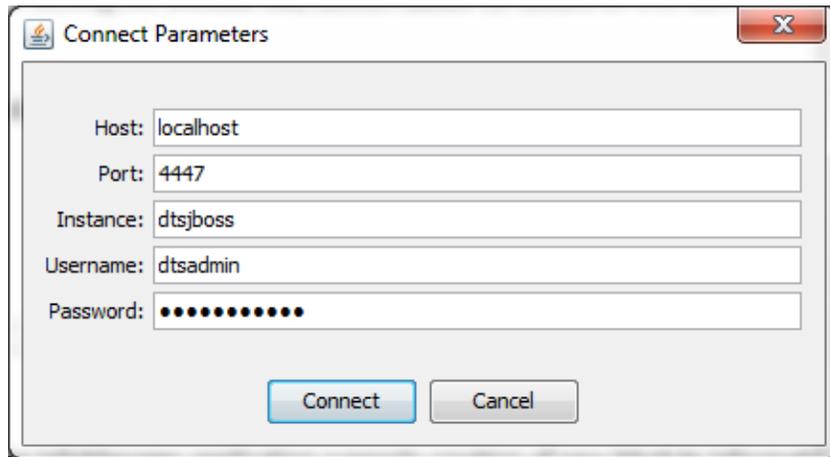
The **Kit name** is the name of the version's installation kit. The file name must end in “.zip”. When a new Version is created, the value of **Kit name** defaults to the Module's jar name (less the extension), a dash character and the version name. This value can be edited as required.

A Version **Description** is required. This is usually a brief list of new features in the Version. This description will be shown in Module Info panel of the Module Manager window. Note that html tags are permitted in the description.

Once any Module (or Version) information has been changed, the action buttons at the bottom of the panel are enabled and an asterisk appears in the InfoManager Title Bar. Click on **Save Module** to save any pending Module changes, or **Restore Module** to discard the changes and reload the original data. If this is a new Module Information File, **Save Module** performs a **Save As ...** operation.

### C.2 Setting the Module URL

To set the Module URL for a DTS server instance, select the **Set Module URL ...** item in the InfoManager **Configure** menu. This opens a standard DTS Connection dialog:



Enter the desired connection parameters, then press **Connect**. The InfoManager connects to the requested DTS server, and shows the Set Module URL dialog.

Enter the new Module URL and click **OK**. This will save the URL in the server's Module URL System Property. Any subsequent DTS Editor invocations of the Module Manager facility will use the Module Folder at this URL.