



APELON

BECAUSE TERMINOLOGY MATTERS

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

+1 (203) 431-2530

www.APELON.com

www.ApelondTS.org

DTS 4: Versioning Guide

Table of Contents

Introduction	3
Versioning Overview	4
Versioning of Namespaces	4
Versioning of Subsets	5
Versioning in the DTS Editor	6
The Version Select Widget.....	6
The Search Panel	7
The Concept/Term Details Panel.....	7
The Concept Compare Panel	7
The Concept History Panel	8
The Subset Compare Panel.....	9
Publishing a New Namespace Version	10
Publishing a New Subset Version	11
Versioning in the DTS API.....	11

Introduction

DTS Version 4 introduces comprehensive version management of both subscription and locally-defined terminologies and associated DTS Subsets. Version management, or simply *versioning*, provides an integrated view of a terminology's evolution over time. DTS versioning supports searching and browsing of a terminology based on both specific *dates* and specific released, a.k.a. published, *versions*. Details of Concept, Term, and Attribute creation and retirement can be determined, and a full audit trail of "who did what to which Attribute" is maintained for locally-defined content. This Guide has been written to assist clinicians, informaticists, and developers understand the basic DTS versioning model and to utilize the DTS Browser, DTS Editor, and API to create, maintain, and deploy versioned terminologies.

The **Versioning Overview** section describes the philosophy of version management in DTS and gives a summary of the methods used to expose versioning functionality in the product. Versioning of both Namespaces and Subsets is presented. This section is required reading for all DTS users.

Versioning in DTS Applications describes how versioning is implemented in the DTS Browser and DTS Editor. While many of the capabilities are similar in the two applications, this section describes the specific features available in each application and how they can be most effectively utilized. It is recommended that all DTS users review this section as it provides a number of detailed examples of version management situations and procedures.

Finally, **Versioning in the DTS API** provides specific technical information on the additions to the base DTS API relevant to version management. A clear understanding of these new features is necessary for any programmer looking to add version management capabilities to existing DTS applications, or to develop new DTS Modules. This section assumes the reader has a basic understanding of Java programming and is already familiar with the "classic" DTS API. See the DTS API Javadoc and DTS API Tutorial for the necessary background information. This section can be skipped by those who will only use the DTS Browser and DTS Editor.

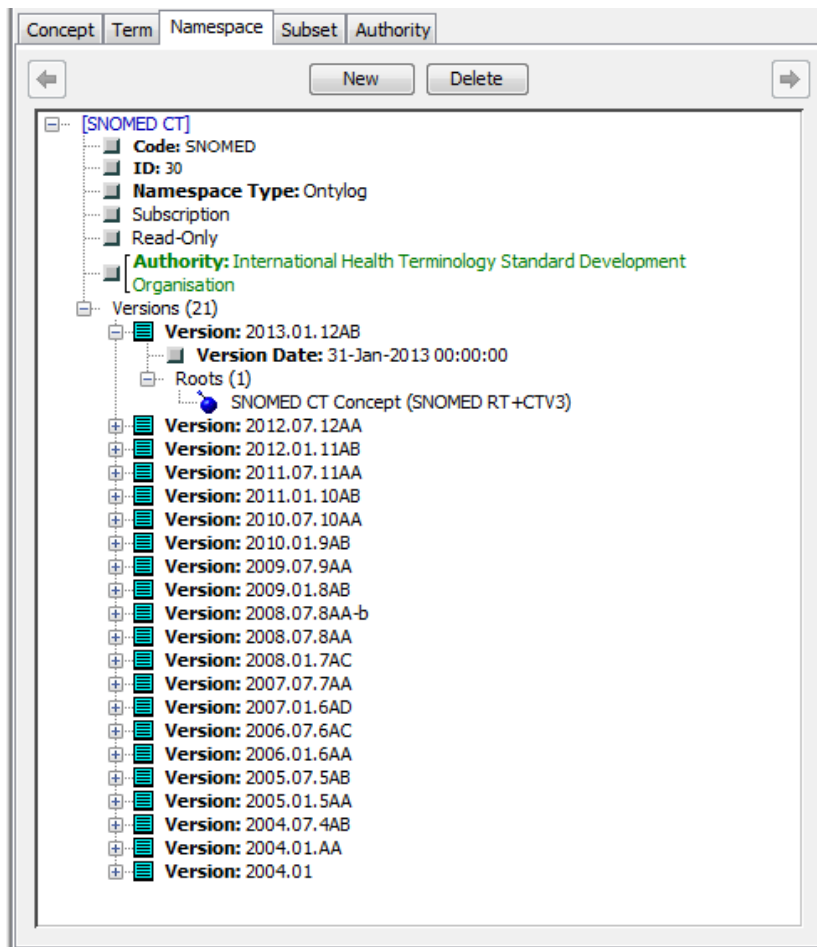
Versioning Overview

Versioning of Namespaces

A **Version** of terminology content is the state of that content at a specified point in time. We generally speak of a piece of content being “published” to create a Version. Once published, the state of the content does not change; i.e., it is immutable. Thus one speaks of the *January 2012 Release of SNOMED CT International* as a Version, as is the *2013 Version of ICD-10-CM*.

DTS deals primarily with terminology content made up of Concepts, Terms and their associated Attributes, for example, Properties and Associations. This content is “packaged” (modeled, presented, and distributed) in management units called **Namespaces**. In DTS 4, the complete history of a single terminology (aka Code System) is maintained in its Namespace. Associated with each Namespace is a set of **ContentVersion** objects that describe the published states (Versions) of the terminology.

The DTS API supports query and browsing relative to specific Namespace Versions and similar features are exposed in the DTS Browser and DTS Editor. The screen shot below shows the DTS Editor Details Panel (Namespace tab) for SNOMED CT.



Every `ContentVersion` object has a Version Name and Release Date. Additional attributes, including Version Properties and Namespace Roots, can also be added to each Version. Version Properties can include other dates, such as Effective Date or Expiration Date, or explanatory text.

For subscription Namespaces, such as SNOMED CT shown above, only formally published Versions are defined and all such Versions have a specific Release Date. For local Namespaces, a special “Working” Version is always present in addition to published Versions. This Working Version does not have a Release Date and additions and edits to Concepts/Terms in the Namespace are always made to this Version. When a Namespace is Published, the data in the Working Version is finalized (see details below), and becomes the stated Version. A new Working Version is then created for future development.

When performing a search in a Version, DTS refers to the returned results as consisting of **Concept Snapshots**. A Concept Snapshot is simply the state of the Concept in the requested Version (at the requested date). The Snapshot consists of the base Concept information (relative to the Version, see additional information below) and all Attributes of the Concept present in that Version. Generally, terminology applications deal exclusively with Concept Snapshots, but DTS Editor panels are available to compare Concept Snapshots between two Versions and display the full Attribute history of a Concept across Versions. See **Versioning in DTS Applications** below for further information.

Modern terminology practice requires that Concepts are never truly “deleted” from a terminology; they are, rather, retired or made inactive. DTS uses the words **Active** and **Inactive** to denote the state of a Concept or Term. Thus, Concepts, Terms and their Attributes have a new attribute called **Status** which can assume the values `Active` or `Inactive`. (A third value, `Deleted`, is available during data modeling of local content. This is a temporary Status, however, and is never present in a published Version.) Functions in DTS allow searching for just `Active` Concepts/Terms (the default) or all Concepts/Terms.

For each Concept, Term and Attribute, DTS also maintains an “audit trail” consisting of the Creation Date, Retired Date, and User associated with each modification. Snapshots of Active Concepts and Terms only show those Attributes that are active (not retired) for the Concept/Term as of the Snapshot Date/Version. Additional functions are available to retrieve the Attribute history of a Concept along with each Attribute’s audit trail information.

Versioning of Subsets

Subsets present their own unique issues with Versioning. As independent DTS objects, Subsets can have Versions (and associated `ContentVersion` objects) similar to Namespaces. But Subsets are also dependent on the Namespaces and Subsets (and their Concepts) upon which the Subsets are defined. Thus Version 1 of the `CommonProcedureSubset` could be built with Concepts from the *January 2012 Version of SNOMED CT International*. A modeler could then create Version 2 of this Subset using the same January 2012 Version of SNOMED, or could choose to have Version 2 of their Subset built on the July 2012 Version. The DTS Subset Editor shows the Versions defined for any Subset.

In order to manage these dependencies, DTS requires that the definition of a Subset specify (in its Namespace and Subset filters) the Version of the Namespace/Subset from which Concepts are drawn. Concepts from different Versions of the same Namespace/Subset are not permitted.

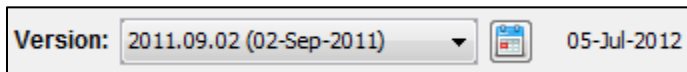
The Subset Editor module in the DTS Editor provides functions for previewing and comparing pending Subset expressions. A Subset Compare option is also available to compare the contents (Concept members) of two different published Versions. Comparison can be made between different Versions of the same Subset, or between versions of two different Subsets. See [Versioning in the DTS Editor](#) for further information.

Versioning in the DTS Editor

The DTS 4 Editor fully supports searching, browsing and manipulating Versioned objects.

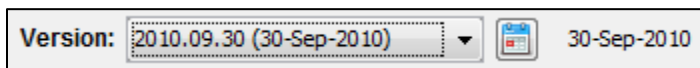
The Version Select Widget

The most obvious addition to the DTS Editor GUI for versioning is the presence of the **Version Select** widget in the Details, Search, Tree, Walker, Concept Compare, Concept History and Subset Compare panels. This widget, shown below, enables the selection of snapshot dates and associated versions.

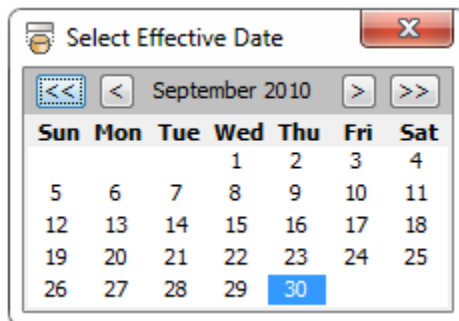


Once a Namespace (or Subset) is selected, in the Tree panel for example, the Version Select widget combo is loaded with the set of associated versions. The versions are sorted in temporal order (most recent first) and the latest is selected. The date at the right of the widget is the widget's effective date, initially the current date.

To view the state of the Namespace Tree for a different version, select the version from the combo. The effective date is set to the date of the version and the Tree is refreshed for this version:



To view by date rather than version, click on the Calendar icon. This brings up the "date picker" window:





Select a date by clicking on a calendar entry, or navigate across time using the previous year (<<), previous month (<), next month (>) or next year (>>) buttons. Once a date is selected, the date picker closes, the effective date is updated to the selected date, and the associated version is shown in the combo. Finally, the Tree (in this case) is refreshed for the associated snapshot date.

The Version Select widget is automatically loaded when Concepts (or Terms) are transferred between panels. Dragging a Concept from the Tree panel (whose effective date is 30-Sep-2010) into the Detail

panel will show the correct Concept Snapshot (from 30-Sep-2010) and initialize the Version Select widget to this version/date.

The Search Panel

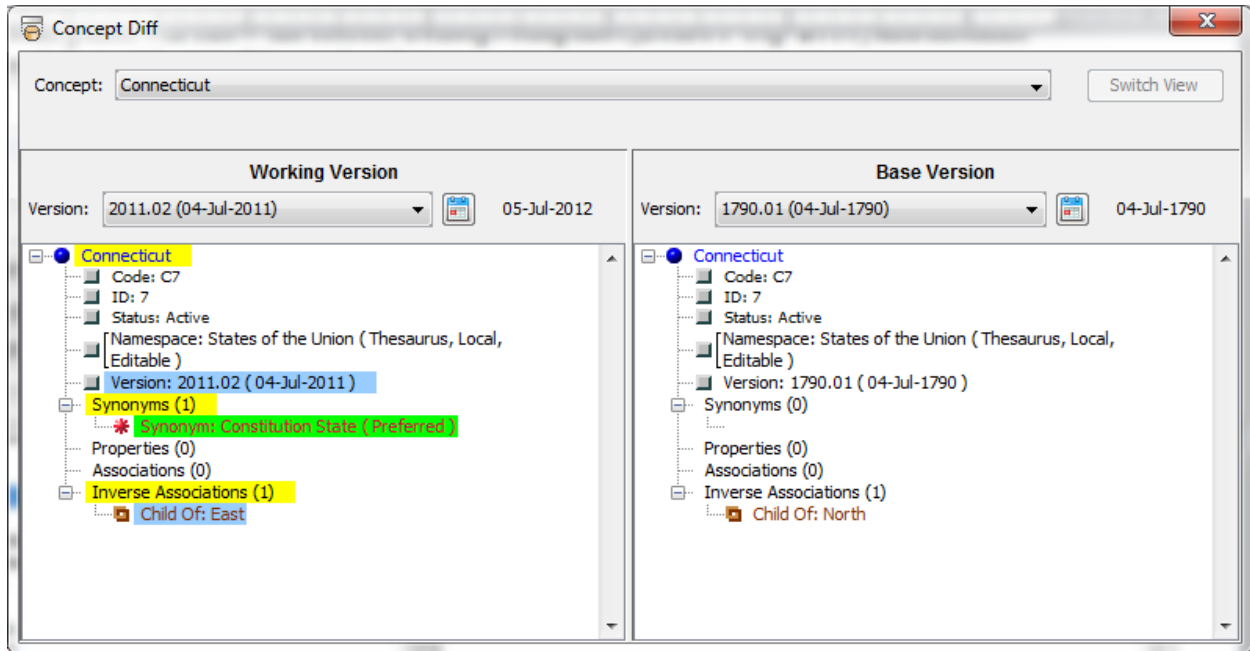
In addition to inclusion of the Version Select widget, the Search panel adds the `Status` combo at the top of the panel. The value of this combo determines the Status (type) of Concepts that will be returned by the search. `All`, `Active`, `Deleted`, and `Inactive` values are available. The default value is `Active` providing consistency with pre-DTS 4 behavior. When `All` is selected, a Status icon (blank for `Active`,  for `Inactive`, or  for `Deleted`) is shown to the left of the returned Concepts.

The Concept/Term Details Panel

Three additions have been made to the Concept and Term tabs of the Details panel for Versioning. First, the Version Select widget has been added to the bottom of the panel. Use this widget to change the snapshot date of the focus Concept. Second, the `Status` and `Version` attribute lines have been added to the display. Presence or absence of these attributes can be controlled by the `Configure View` popup panel. Finally, similar to the DTS Browser, attribute history (`Added Date`, `Retired Date` and `User`) can be shown as `ToolTips` on the elements. Showing of `History ToolTips` is also controlled by a checkbox in the `Configure View` popup panel.

The Concept Compare Panel

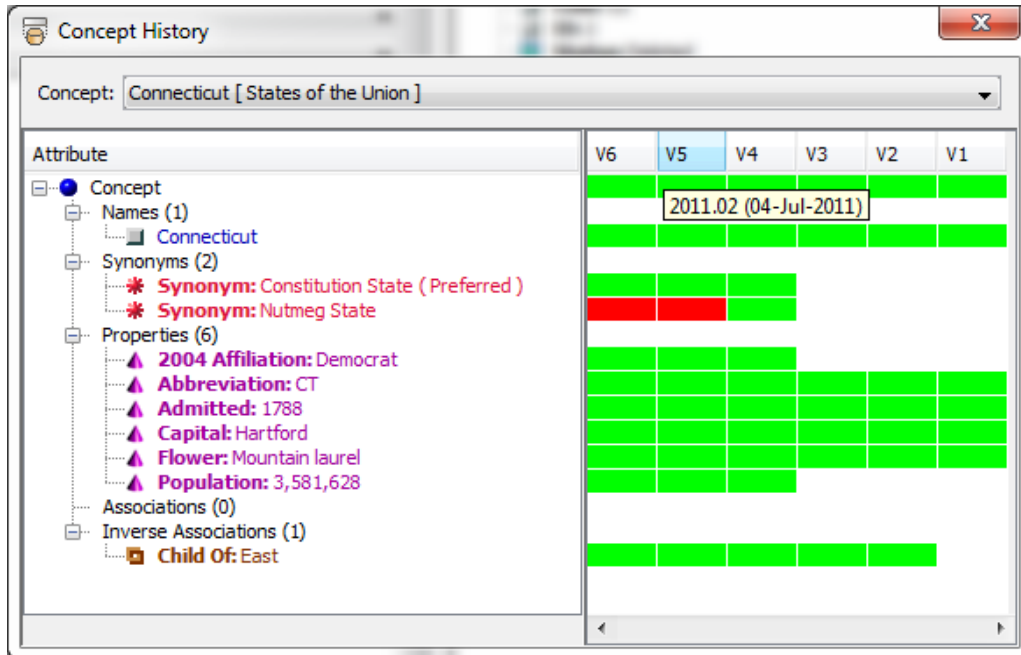
The Concept Compare panel, available in the `Tools` menu, provides a visual comparison of two versions of a Concept. Drag a Concept from any other panel into the `Concept` combo (the combo maintains a history of the loaded Concepts). The left (`Working Version`) panel will be loaded with the dropped snapshot of the Concept. The right (`Base Version`) panel will be loaded with a snapshot of the Concept from the immediately previous version of the Namespace (note the Concept may not have existed in this Version). Differences between the Versions are shown in color highlights: yellow for a change in subordinate attributes, green for an added attribute, blue for a modified attribute and red (not shown in the below) for a deleted attribute.



The two Version Select widgets can be used to change the Working and Base version, with the restriction that the base version must not be later (more recent) than the working version.

The Concept History Panel

The Concept History panel, also available in the **Tools** menu, provides a representation of the complete history of the Concept across all Versions. Drag a Concept into the Concept combo box to load the panel. The left side of the display shows the tree representation of the Concept, and includes all attributes of the Concept that have ever been created. The right side of the display shows the status of each Concept Attribute in each Version. Active Attributes are shown in green, deleted Attributes in red. The oldest Version (shown as “V1”) is to the far right of the display. The most recent version is to the far left. ToolTips on the Version column headers show the actual Version name and Release Date, while ToolTips on the colored cells show the revision details of the Attribute for that cell/Version.

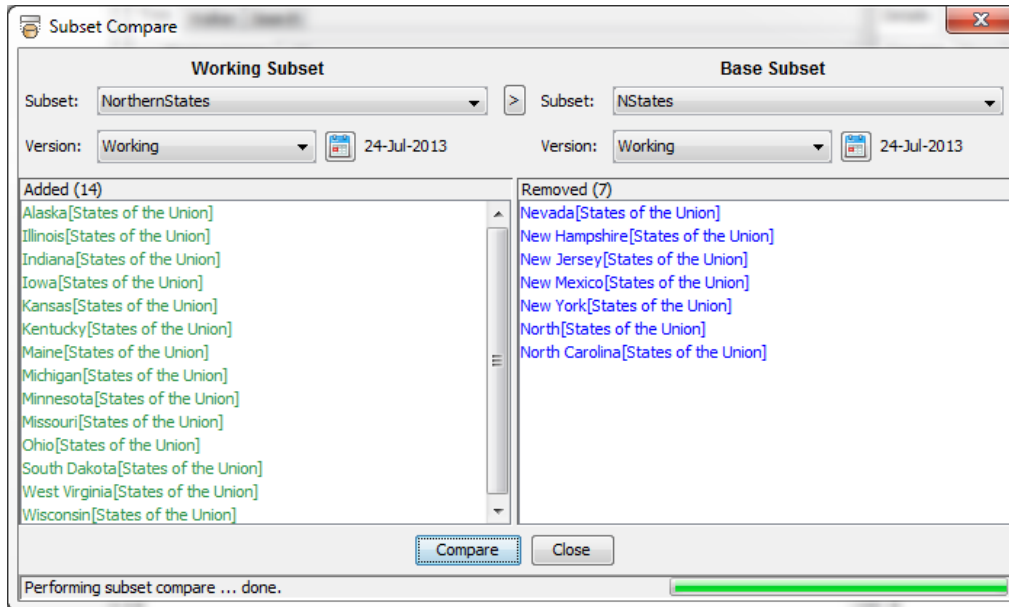


The Subset Compare Panel

To compare the Concept membership between two Versions of the same Subset, or between Versions of two different Subsets, select the Subset Compare panel, available in the **Tools** menu. To compare two Versions of the same Subset, select the desired Subset from the left (“working”) Subset combo. Select the desired Version from the Version Select widget. Then press the copy button (“>”) to automatically select the same Subset in the base Subset combo and select the desired base Version. (You can also just select the Subset from the base combo.) Finally, press **Compare** to display the differences. Concepts that are in the working Version and not in the base (“added” Concepts) are shown in the left half of the results panel (see example below). Concepts that are in the base Version and not in the working (“removed” Concepts) are shown in the right side.

To compare Versions of two different Subsets, one Subset from the left Subset combo, and the other from the right combo. Then select the desired Versions, and click **Compare**.

DTS 4: Versioning Guide

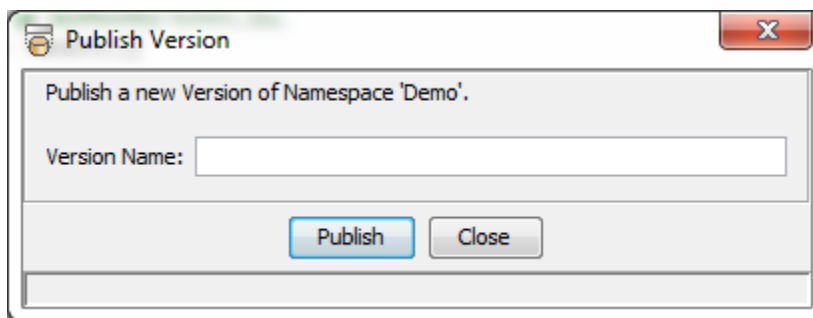


Publishing a New Namespace Version

The Working Version of a local Namespace can be published using the Publish function in the Namespace Editor pane. Note that Publish is only available from the Namespace Editor, and not the visually similar Namespace tab of the Details panel.

In order to Publish a Namespace, the current User must have the *Manage* Permission on the Namespace and the Namespace must be Writable. See the ***DTS 4.0 User Manager Guide*** for further information on User Permissions. If the Namespace is an Ontylog Extension Namespace, additionally, the classification state of the Namespace must be current.

After selecting the desired Namespace in the panel, click on the Publish button. This opens the Publish Dialog:



Enter the Name of the new Version and click Publish. The following steps are then performed:

DTS 4: Versioning Guide

1. The contents of the current Working Version are “finalized”. Finalization permanently removes all interim, i.e., inactive, Attributes and any Concepts with the Deleted Status. Concepts with Inactive Status are retained. Only the latest Attributes remain in the Version after Publishing.
2. The Working Version is given the designated Version Name and its Release Date is set to the current date.
3. The new Version is set to read-only.
4. A new Working Version is created.

Publishing a New Subset Version

Publishing the Working Version of a Subset is similar to that of a Namespace. In order to Publish a Subset, the current User must have the *Manage* Permission on the Subset and the Subset must be Writable. See the DTS User Manager Guide for further information on User Permissions. Additionally, the build state of the Subset must be current, i.e., the Subset Expression cannot have changed since the most recent Subset build.

Open the Subset Editor, select the Subset to be published, and click on Publish. This opens the Publish Version dialog as above. The publishing steps performed are:

1. The contents of the current Working Version are “finalized”. Finalization permanently removes all interim, i.e., inactive, Attributes and any Concepts or Terms with the Deleted Status. Concepts and Terms with Inactive Status are retained. Only the latest Attributes remain in the Version after Publishing.
2. The new Version is set to read-only.
3. A new Working Version is created.

Versioning in the DTS API

Both the `NamespaceQuery` and `SubsetQuery` classes implement methods to enable handling of versions:

```
ContentVersion findVersionById(int id, int contentId,  
VersionAttributeSetDecsriptor asd)  
  
ContentVersion findVersionByName(String name, int contentId,  
VersionAttributeSetDescriptor asd)  
  
ContentVersion[] getVersions(int contentId)  
  
ContentVersion getContentVersion(int contentId)  
  
ContentVersion getContentVersion(int contentId, Date date)
```

The first two methods retrieve a specific `ContentVersion` by its id or name given the associated `contentId` (Namespace Id or Subset Id). The third method retrieves all the `ContentVersions` defined for the specified Namespace/Subset. The last two methods return the most recent `ContentVersion` (date is “now”) or the `ContentVersion` whose release date is less than or equal to the specified date, i.e.

the “current version” for that date. These methods are typically used by browsing and searching applications.

As described in the **Versioning Overview** section, `Concept` and `Term` objects in DTS 4 (`DTSConcept`, `OntologConcept` and `Term`) are really Snapshots. Retrieval of specific `Concept` (and `Term`) Snapshots is accomplished by additions to the `ConceptAttributeSetDescriptor` (and `TermAttributeSetDescriptor`) classes. Prior to DTS 4, these classes were used to “filter” the attributes returned on `Concept` objects from many search and navigation query methods. Attribute set descriptors in DTS 4 have been extended to include a `SnapshotDate` attribute. By executing `setSnapshotDate(date)` on a attribute set descriptor, returned `Concepts`’ Snapshots will be relative to that date. To support backwards compatibility, the default snapshot date for attribute set descriptors is the current date, thus ensuring that `Concept` Snapshots are from the most recent version.

`Concept` Snapshots carry their effective dates with them. The `DTSConcept.getSnapshotDate()` method will return this date, so it is always possible to determine the associated date for a `Concept` Snapshot. This enables consistent movement of Snapshots within applications, such as when hyperlinking `Concepts` in the DTS Browser or dragging and dropping `Concepts` in the DTS Editor.

It is also possible to search for `Concepts` (and `Terms`) that have a specific `Status`. The mechanism is a new attribute in the `DTSSearchOptions` class. Use the `DTSSearchOptions.setStatus(ItemStatus status)` method to restrict search results to `Concepts` having the specific `Status`. A search `Status` of `null` (the default) will return all `Concepts`, independent of `Status`. The `Status` of individual `Concepts` can, of course, be found from `DTSConcept.getStatus()`.

For further information on versioning in the DTS API see the DTS Programming Tutorial in the Javadoc.