

# Realm Codes and Release Numbers in V3 Artifact IDs

GW Beeler  
10/27/2006

## 1 Background & History

It has been the "design strategy" of those responsible for tooling to have the databases (both PubDb and Access Design Repository) contain only the base ID (without a realm), or at most a 15-character code, with the default realm being "UV" for the shorter IDs.

### 1.1 Release numbers

Further, we have intentionally **not** placed release numbers (the "nn" or "01" extensions) in the databases at all. Rather, these are specified in another table of the database (HL7\_artifactBallotStatus) and are added by RoseTree when the data are processed. As noted above this same pattern and approach is used in both the PubDb and the Design Repository, and RoseTree is used to process both of these data sources to xml.

The reason for this decision is simple. Release numbers are not known until after a specification becomes standard (either as full normative or DSTU). Moreover, they are not under the control of the Technical Committees, but rather are determined by HQ as specifications get presented to, and approved by ANSI. Finally, most of the specifications use CMETs which have gone through four releases by now. Thus, it seemed wise to maintain control of release number tracking in a central DB table that could be readily updated in individual PubDbs and Repositories as they changed.

Note: the HL7\_ArtifactBallotStatus table **also** provides the Icons used in publishing to distinguish Committee ballot from Membership, Normative from DSTU, etc.

### 1.2 Realm codes

Until recently, we had decided to exclude the "UV" realm designation from all databases in order to assure consistency. (In retrospect, it would have been better to decide to **include** the "UV" realm designation everywhere, but that is the advantage of hindsight.) In the last few cycles, we have had selected Canadian artifacts in the ballot that were not yet normative. Thus, these appeared with a realm code of CA, but no release numbers and had to be edited by hand in some cases.

## 2 Current Changes

The current status was challenged by M&M last winter, and we agreed to move to an arrangement in which **all identifiers will have a realm code**. This change is being implemented this cycle.

### 2.1 Support in PubDb and DesignRepository

To support this, releases of the PubDb and DesignRepository now include a "Realm Code Management Widget" that facilitates correcting all identifiers within a DB by assigning

them a realm code. ("UV" will be the default, but this can be changed by the user.) Indeed, when the DB is first opened in Access, it checks to see if any artifacts have a faulty id (no realm, or include release numbers) and if any are found, the widget is automatically opened to allow these to be corrected.

The same widget supports a search/replace capability **but** this can only be applied to all identifiers in a given table (such as MessageType). More fine-grained changes will have to be made in the database itself.

## **2.2 Support in PubDb Manager**

The PubDb Manager "widget" distributed as part of the publishing tools also includes capability to do artifact id "**corrections**" (cases where there is no realm code or a release number exists) as a process on an entire database, or in a batch on a set of databases. This capability allows the selection of the realm to be assigned in this correction, but provides no search/replace capability. **Note:** HL7 publishing staff will use this capability to correct faulty IDs and assign them to "UV" for any databases submitted for publication that have such flawed IDs.

## **2.3 Support in RoseTree<sup>1</sup>**

### **2.3.1 Recent releases**

Since RoseTree's primary focus is on developing and publishing the standards, various quality assurance (QA) tasks have been encoded in it. Among these is the task of "cleaning up" and aligning artifact IDs from the database. Thus, in the recent past, if it loaded an ID from the DB that contained a release number, it automatically removed the realm code and release number, leaving the raw "base" identifier.

RoseTree also performs the matches to the ballot artifact status table (HL7\_artifactBallotStatus) that are used to specify the release level of a normative or DSTU product.

### **2.3.2 New release**

Effective with release 4.02 of RoseTree, this support changes as follows.

- 1) When loading a model from a repository, RoseTree:
  - a) Strips to IDs to 15 long (includes realm) rather than 13, by default
  - b) Still strips to 13 (base ID) when the source has 'nn' attached. The 'nn' **should** be coming from the ballot artifact status table
  - c) There is an escape option for the behavior in (2) when loading either a single-HMD or a batch of them. (See section on inhibiting stripping, below)

---

<sup>1</sup> RoseTree is an **ancient** utility that was first released a decade ago. It has been modified, as needed, to support HL7 development. Although the Tooling Committee **hopes** to replace its functionality with open source tools based on HL7 MIF files, that end has been slow in coming. Hence we are all dependent on a single tool developed and maintained by a single individual mostly in his spare "volunteer" time. As changes have been made, they have focused on the need to support standards development and publishing. Other uses, such as implementation support have been accommodated as time and brain energy permit.

- d) If it finds an ID that is 13 characters (either originally, or as a result of stripping from (2) above), it will assign the "**UV**" domain by default. There is no planned RoseTree capability to allow this default to be changed. Other tools will allow one to edit these IDs in bulk within the repository.
- 2) When matching an artifact ID to the Ballot artifact status tables, it seeks a match in the following order, and uses the first match it finds:
    - a) Match to the first 15 characters (like POLB\_MT112233CA) [preferred]
    - b) Match to the first 13 characters (like POLB\_MT112233) [to support old status tables]
    - c) Match to the full, original ID, whatever that may be
    - d) If no match, it then proceeds to use inheritance (from Domain & topic)

### 3 Managing Unusual Realm/Release IDs

At times, implementers may face the task of working with unusual design identifiers. Recently, I was asked about problems users had when dealing with an artifact whose identifier was "PORT\_HD050001HT01". The problem was that RoseTree saw this as invalid (due to the "01" release number) and was stripping this back to 13 characters. From there, it proceeded to assign a realm of UV, and a release of "01" (because the RT specification is at release 1.)

In the future, there are two ways to address this problem – using the ballot status table, or inhibiting RoseTree's QA function..

#### 3.1 Assigning Release Numbers through Ballot Artifact Status

The preferred approach is to follow the pattern used in publishing the specifications. That is, create the design with a 15-character ID, "PORT\_HD050001HT" in the example above, **and** create an entry in the HL7\_artifactBallotStatusTable for the specific artifact ("PORT\_HD050001HT") and specify the status as "Normative Standard" (or "DSTU") and the release number as "01". With these settings, RoseTree will create the desired result using all default settings.

#### 3.2 Inhibiting Correction of Ids When Loading

The alternative approach is to represent the full ID, including the release number, in the design, when it is passed from Visio through RoseTree to the repository, and then "Inhibiting" the ID correction functions when loading the designs and when exporting them to xml (for "hmd" files).

In order to inhibit correction, do one of:

- o When loading a single model in RoseTree, check the "**Inhibit ID correction**" box at the bottom of the model-open dialog box, near the "OK" button.
- o When exporting all designs from a single database, select the "**NO stripping of ID**" radio button in the "**Strip IDs of UVnn**" frame, on the xml-output options select page.

- When preparing to batch output from multiple databases, select the "*Not Strip IDs*" option from the menu **File:Manage model files...Batch HMD output...**

The disadvantage of this method (other than it is not the preferred option) is that the choice to inhibit correction must be made each time a file is opened or exported.