Programmer's Guide to the UID Facility

Subroutines for Generating Unique Identifiers

Stephen M. Moore

Mallinckrodt Institute of Radiology
Electronic Radiology Laboratory
510 South Kingshighway Boulevard
St. Louis, Missouri 63110
314/362-6965 (Voice)
314/362-6971 (FAX)

Version 2.10.0

August 3, 1998

This document describes a simple facility for generating
unique identifiers (UIDs) for use by DICOM applications.

1 Introduction

The DICOM V3 Standard provides mechanisms for labeling items (such as Information Objects, SOP Classes) with unique identifiers (UIDs). NEMA (through its Committees) defines abstract elements required for the V3 Standard and assigns unique identifiers to these elements. Examples of these UIDs are the Transfer Syntax UIDs defined in Part 5 of the Standard.

Devices that create new objects (patient records, image series, image instances) are required to generate UIDs using a method based on the OSI object identification defined by ISO 8824. Part 5 of the DICOM Standard provides an example that can be used for generating UIDs, but this method is not a requirement of the Standard. This facility provides a set of routines for generating unique identifiers using a scheme which is similar to the example provided in Part 5. Each UID generated by this facility is of the form:

root.deviceType.serialNumber.dataType.suffix

root — a numeric prefix based on a value assigned by the appropriate standards committee. In our case, this root identifies Washington University/RSNA demo.

deviceType — a single numeric field which identifies the device type. A list of device types is included below.

serialNumber — a single numeric field which identifies the serial number of the device. This number uniquely identifies the device.

data type — a single numeric field which identifies the type of data. A list of data types is included below.

TABLE 1. Device Types

1	MR Instrument
2	CT Instrument
3	CR Instrument
4	US Instrument
5	NM Instrument
6	Film Digitizer
7	Video Capture Unit
8	General Secondary Capture Unit
9	Imaging Workstation
10	Database

TABLE 2. Data Types

2	Patient
3	Visit
4	Study
5	Series
6	Image
7	Results
8	Interpretation
9	Printer

The *suffix* value contains a single numeric field. This facility provides one function which generates UIDs for any of the data type defined. It maintains unique numbers for each data type in an ASCII file. The name of the file is read from the environment variable UIDFILE. Each line in the ASCII file contains a keyword followed by an integer. The facility generates the next unique number by incrementing the appropriate integer and rewriting the entire file. Keywords are defined below:

ROOT	Root value for this device
DEVICE	Device type
SERIAL	Serial number
PATIENT	
VISIT	
STUDY	
SERIES	
IMAGE	
RESULTS	
INTERPRETATION	
PRINTER	

An example of a “UIDFILE” is shown in Figure 1 below. This file describes a CT scanner. Note that there are entries in the file for items that a CT scanner will not create (such as interpretation)

ROOT	1.2
DEVICE	9
SERIAL	1
PATIENT	5
VISIT	1
STUDY	2
SERIES	5
IMAGE	101
RESULTS	1
INTERPRETATION	1
PRINTER	1

FIGURE 1. Example “UIDFILE”

This facility also provides a function which allows the caller to ask for the next available integer number for a data type. Once this number is taken from the UIDFILE, it will not be reused to make a unique identifier.

The include file for this facility also includes constant definitions for all of the UIDs which are found in the DICOM Standard.

Important Note: The example UIDFILE above contains an illegal root value. Roots are assigned by appropriate standards committees, and the root used for the example above was not assigned to this organization. It is used only as an example. The algorithm used by this facility to generate UIDs was designed mainly for the purposes of the RSNA `93 DICOM demonstration. There are much better methods for generating UIDs that vendors should investigate for use in commercial equipment.

2 Data Structures

There are no publicly accessible data structures.

3 Include Files

To use UID functions, applications need to include these files in the order given below:

```
#include “dicom.h”
#include “dicom_uids.h”
```

4 Return Values

The following returns are possible from the UID facility:

UID_NORMAL	Normal return from UID routine.
UID_NOUIDFILENAME	UID function could not translate UIDFILE environment variable into a legal file name.
UID_GENERATEFAILED	UID function failed to generate a unique identifier or a unique number.
UID_FILEOPENFAILURE	UID function failed to open UID file for read.
UID_FILECREATEFAILURE	UID function failed to create UID file for write.
UID_ILLEGALROOT	UID function failed to read root value from UID file.
UID_ILLEGALNUMERIC	UID function failed to read a numeric value from UID file.
UID_NODEVICETYPE	UID file is missing device type field.
UID_NOROOT	UID file is missing root field.

5 UID Routines

This section provides detailed documentation for each UID facility routine.

UID_NewNumber

Name

UID_NewNumber - generate a new number for a study or series.

Synopsis

CONDITION UID_NewNumber(UID_TYPE type, unsigned long *value)

type	One of enumerated UID_TYPEs (UID_STUDY, UID_SERIES). This parameter defines the type of number to be returned.
value	Address of variable to hold the new number.

Description

UID_NewNumber generates a new integer which may be used to label a DICOM item such as a series or a study. This function reads the UIDFILE and finds the next available number appropriate to the caller's type argument. The next available number is stored at the address specified by the caller (**value*), and the updated file is rewritten.

Notes

This function assumes the environment variable UIDFILE contains the name of a file which contains the UID information. This file name can be a full or relative path name. This function must have both read and write access to the file, since the file is rewritten after each new number is generated.

Return Values

```
UID_NORMAL
UID_GENERATEFAILED
```

UID_NewUID

Name

UID_NewUID - generate a new, unique identifier.

Synopsis

CONDITION UID_NewUID(UID_TYPE type, char *uid)

type One of enumerated UID_TYPEs (UID_STUDY, UID_SERIES). This parameter defines the type of UID to be returned.
uid Address of memory allocated by caller to store generated unique identifier.

Description

UID_NewUID generates a new UID which may be used to label a DICOM item such as a series or a study. This function reads the UIDFILE and finds the next available number appropriate to the caller's type argument. This number is then used in combination with other values in the UID file to create a unique identifier. The newly created UID is stored at the address specified by the caller (**uid*), and the updated file is rewritten.

Notes

This function assumes the environment variable UIDFILE contains the name of a file which contains the UID information. This file name can be a full or relative path name. This function must have both read and write access to the file, since the file is rewritten after each new number is generated.

Return Values

UID_NORMAL
UID_GENERATEFAILED