Programmer's Guide to the Print Facility

Support Routines for Access to PRN Structures

Aniruddha S. Gokhale

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

The PRN Facility performs two services for programmers working with the following Service Class defined in DICOM V3.0: Print Management Service Class. The first service builds a DICOM object suitable for network transmission. The second parses a DICOM object so that its attributes can be used.

1 Introduction

The PRN facility provides support operations for two DICOM V3.0 Meta SOP Classes. These SOP Classes are defined in Part 4 of the Standard and are:

- Basic Greyscale Print Management Meta SOP Class
- Basic Color Print Management Meta SOP Class

Part 4 defines the commands that are valid for the various print management SOP classes and the attribute lists that are required for each command. Typical commands are N-CREATE, N-SET and N-DELETE. The PRN facility defines a fixed structure for each type of attribute list that is defined in the print management section of Part 4. There is typically one structure defined for each print management SOP class. The PRN facility provides routines which translate the fixed structures to and from DCM objects. The DCM objects can then be used by other facilities to send complete messages to peer applications. This facility does not define any of the logic necessary to implement the print management classes. It merely provides support for the necessary attribute lists (which we think of as information object definitions).

Each structure has the following elements: all the attributes for that SOP Class defined in Part 4, an instance UID (unique identifier), a SOP Class UID, a PRN specific type and an attribute flag. The attribute elements of the structure are used to hold attribute values. A reserved field is always the first field in all the structures and is used by the LST facility. This field is followed by the PRN type field used to distinguish various structures. The UID elements are not translated by this facility but are included as placeholders to be used by other facilities and applications. The flag variable element of the structure holds up to 32 flags that are set when a corresponding attribute element is valid. All structures are of one of the PRN_XXX types where the XXX can be any one of :

- Basic Film Session
- Basic Film Box
- Basic Image Box
- Basic Annotation Box
- Basic Print Job
- Printer

The first function associated with each Service Class, called PRN_BuildObject, builds a DICOM object from a PRN_XXX structure. PRN_BuildObject knows which attribute elements of the structure to place in the object from the flag variable. If a bit in the flag variable is set, the value in the corresponding attribute element will be placed in the object.

The second function associated with each Service Class, PRN_ParseObject, parses a DICOM object into a PRN_XXX structure corresponding to the PRN type parameter to this function. Only those attributes associated with the previously mentioned Service Classes are parsed from the object. This function also sets the bits in the flag variable corresponding to what attributes were found.

2 Data Structures

The PRN_XXX structures are defined in the dicom_print.h include file. The PRN_BASICFILMSESSION structure is presented below as an example.

```
typedef struct{
    void    *reserved[2];
    PRN_TYPE  type;
    long    bfsAttributeFlag;
    char    filmSessionSOPClassUID[DICOM_UI_LENGTH+1];
    char    fileSessionSOPInstanceUID[DICOM_UI_LENGTH+1];
    char    copies[DICOM_IS_LENGTH + 1];
    char    priority[DICOM_CS_LENGTH + 1];
    char    mediumType[DICOM_CS_LENGTH + 1];
    char    filmDestination[DICOM_CS_LENGTH + 1];
    char    filmSessionLabel[DICOM_LO_LENGTH + 1];
    char    memoryAllocation[DICOM_IS_LENGTH + 1];
    LST_HEAD  *referencedBFBSeq;
} PRN_BASICFILMSESSION;
```

The flags used to set bits in bfsAttributeFlag are also defined in dicom_print.h. For instance, the flag for Basic Film Session’s copies attribute is defined below

```
#define PRN_BFS_K_COPIES        0x1
```

The include file for the PRN facility, dicom_print.h, contains enumerated types and defined constants used to differentiate amongst each Service Class and to specify the mandatory attributes of each service class. The various PRN types defined by the PRN facility are :

```
PRN_K_BASICFILMSESSION
PRN_K_BASICFILMBOX
PRN_K_BASICIMAGEBOX
PRN_K_BASICANNOTATIONBOX
PRN_K_BASICPRINTJOB
PRN_K_BASICPRINTER
PRN_K_BASICGREYSCALEIMAGEBOX
PRN_K_BASICCOLORIMAGEBOX
```

Each SOP Class has mandatory attributes. Thus, the appropriate flags in the flag variable must be set (PRN_BuildObject checks for this). To facilitate this, constants that hold the mandatory attribute flags for each SOP Class were defined. An example is given below for a Basic Film Session Service Provider for the N-CREATE DIMSE service:

```
#define REQUIRED_BFS_SCP_CREATEATTRIB   \
 PRN_BFS_K_COPIES | PRN_BFS_K_PRIORITY | \
 PRN_BFS_K_MEDIUMTYPE | PRN_BFS_K_FILMDESTINATION
```

3 Include Files

Any source code that uses any PRN constants, type definitions or functions should include the following files in the order given:

```
#include “dicom.h”
#include “condition.h”
#include “lst.h”
#include “dicom_objects.h”
#include “dicom_uids.h”
#include “dicom_sq.h”
#include “dicom_print.h”
```

4 Return Values

The following returns are possible from the MUT facility:

PRN_NORMAL	Function completed normally
PRN_BUILDFAILED	Build Function failed
PRN_PARSEFAILED	Parse function failed
PRN_LISTCREATEFAILURE	Error in creating a list using the LST facility
PRN_MALLOCFAILURE	Memory allocation error
PRN_ILLEGALUIDLENGTH	Length of UID in error
PRN_LISTFAILURE	A failure occurred using the LST facility

5 PRN Routines

This section provides detailed documentation for each PRN facility routine.

PRN_BuildObject

Name

PRN_BuildObject - build a DCM_OBJECT from a PRN specific structure.

Synopsis

CONDITION PRN_BuildObject(void *structure, DCM_OBJECT **object)

structure	One of the various PRN structures
object	The existing DCM_OBJECT that is to be built.

Description

PRN_BuildObject takes the pointer to the structure as an argument and finds its PRN type. Using this information, it searches for the correct entry in the PRN film table that gives the information about the various attributes and flags of the structure in hand. Using this information, this function then builds a DICOM object using various DCM facility routines.

Notes

If a bit is set in the flag field of the structure and the corresponding structure element does not contain valid data, the DCM_OBJECT will contain invalid data.

Return Values

```
PRN_BUILDFAILED
PRN_ILLEGALUIDLENGTH
PRN_LISTCREATEFAILURE
PRN_LISTFAILURE
PRN_MALLOCFAILURE
PRN_NORMAL
```

PRN_ParseObject

Name

PRN_ParseObject - extracts the attributes associated with the specific SOP Class from a DCM_OBJECT.

Synopsis

CONDITION PRN_ParseObject(DCM_OBJECT **object, PRN_TYPE type, void **structure)

object	Existing DCM_OBJECT to be parsed.
type	The PRN type that is to be parsed from the object
structure	The PRN structure that is to be created and filled in.

Description

PRN_ParseObject takes in a DICOM object and a PRN type as input parameters and constructs the appropriate PRN structure as defined by the PRN type field. The type is searched in the film table and if found, appropriate amount of storage is allocated. As the object is parsed, the attribute flag is set for every attribute that was found in the object.

Return Values

```
PRN_MALLOCFAILURE
PRN_NORMAL
PRN_PARSEFAILED
```

PRN_DumpAttributes

Name

PRN_DumpAttributes - dump an ASCII description of a print strucuture to a file

Synopsis

void PRN_DumpAttributes(void *attrib, FILE *f)

attrib	Pointer to one of the PRN attributes.
f	File pointer to an open file.

Description

PRN_DumpAttributes dumps a description of a PRN structure to a file that was previously opened by the caller. The description includes the name of the field and the value.

Notes

This function can be used to dump a description to stderr or stdout.

Return Values

None