

Programmer's Guide to the Utility Facility

A Facility for Manipulating DICOM Dates and Times

David E. Beecher
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

Copyright (c) 1995, 1998 RSNA, Washington University

1 Introduction

The Utility Facility (hereafter referred to as UTL) contains a collection of routines concerned mostly with manipulating and comparing dates and times in DICOM standard format. All routines are quite simple, but were created to fill a need in the IDB facility.

2 Data Structures

No special data structures are introduced in this facility.

3 Include Files

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

```
#include "dicom.h"
#include "utility.h"
```

4 Return Values

The following returns are possible from the UTL facility:

UTL_NORMAL	Successful
UTL_UNIMPLEMENTED	This function is unimplemented
UTL_MATCH	The regular expression generated a match in the input string
UTL_NOMATCH	The regular expression did not generate a match in the input string

5 UTL Routines

This section provides detailed documentation for each UTL facility routine.

UTL_ConvertDatetoLong

Name

UTL_ConvertDatetoLong - convert a date in DICOM string format to a long.

Synopsis

```
long UTL_ConvertDatetoLong(char *date)
```

date The DICOM date to convert.

Description

Assumes a DICOM date format of “yyymmdd” and converts to a long by pulling out the year and multiplying by 10000, adding that to the month which is multiplied by 100, and adding all that to the day. This is the form of the data retained in the database and is more efficient for comparison purposes than the string equivalent representation.

Notes

None.

Return Values

Returns a long value which contains the functionally equivalent representation of the DICOM date.

UTL_ConvertFloattoTime

Name

UTL_ConvertFloattoTime -This function covers a floating point to a DICOM time.

Synopsis

```
void UTL_ConvertFloattoTime(double dt, char *time)
```

dt The floating point value to convert.

time Holds the equivalent DICOM time string after conversion.

Description

When the DICOM time is converted to a float (actually a double) with the routine *UTL_ConvertTimetoFloat*, it is converted to some number of seconds with the fractional part being significant. Therefore, for this conversion, the variable *dt* is simply converted into hours, minutes, seconds, and fractional seconds in the standard way and then properly formatted and placed into the string pointed to by *time*. The format string is: "hhmmss.ffffff".

Notes

It is assumed that *time* is large enough to hold the formatted string. This routine does not support any alternate time formats.

Return Values

None.

UTL_ConvertLongtoDate

Name

UTL_ConvertLongtoDate -This function converts a long to a DICOM date.

Synopsis

```
void UTL_ConvertLongtoDate(long ld, char *date)
```

<i>ld</i>	The long version of the date to be converted.
<i>date</i>	Holds the converted date value.

Description

The inverse of *UTL_ConvertDatetoLong*, it takes the long, divides by 10000 to extract the year, 100 to extract the month, and whatever is left over must be the day. It is then formatted according to the following DICOM template: “yyymmdd”.

Notes

No alternate date formats are supported.

Return Values

None.

UTL_ConvertRegex

Name

UTL_ConvertRegex -converts a Dicom “regular expression” to the proper regex semantics under UNIX.

Synopsis

```
char *UTL_ConvertRegex(char *regex)
```

regex The DICOM regular expression to convert.

Description

This function converts a DICOM “regular expression” to the proper regex semantics under UNIX. DICOM has only 2 meta characters, “*” to match 0 or more occurrences, and “?” to match a single character. The “*” must be converted to “.*” for regex while the “?” must be converted to “?”. Other special characters to regex like “[“, “[”, and “.” must also be escaped with the “\” character. The DICOM escape character is assumed to be “\”.

Notes

The routine needs to return a string of unknown length. The caller of *UTL_ConvertRegex* should free the string returned after it is no longer needed.

Return Values

A string which is the converted DICOM expression that regex can now understand.

UTL_ConvertTimetoFloat

Name

UTL_ConvertTimetoFloat -convert a DICOM time to a floating point representation

Synopsis

```
double UTL_ConvertTimetoFloat( char *time )
```

time The DICOM time to convert

Description

This function is the inverse of *UTL_ConvertFloattoTime*. It accepts a DICOM time of the form “hhmmss.ffffff” and converts it to an equivalent number of seconds with fractional seconds. The fractional sub-field “ffffff”, may be from 1 to 6 characters.

Notes

No alternate time formats are supported.

Return Values

A double is returned that represents the total number of seconds represented by the DICOM time passed.

UTL_DateMatch

Name

UTL_DateMatch -match a date range as specified in the DICOM standard.

Synopsis

```
CONDITION UTL_DateMatch( char *datestring, char *stm )
```

<i>datestring</i>	The range to be matched.
<i>stm</i>	The date to be compared against the above range.

Description

This routine was written specifically to match a DICOM date string (*stm*) against a Dicom date range (*datestring*). This routine handles the following ranges: <date>, <date>-, -<date>, and <date1>-<date2>.

Notes

None.

Return Values

UTL_MATCH
UTL_NOMATCH

UTL_DeltaTime

Name

UTL_DeltaTime -compute an elapsed time in seconds

Synopsis

```
double UTL_DeltaTime( void *timeStamp )
```

timeStamp The timestamp value returned by *UTL_GetTimeStamp*

Description

UTL_DeltaTime computes the time elapsed from the current time and the time stored in *timeStamp*. The units are seconds.

Notes

Use *UTL_ReleaseTimeStamp* to release the *timeStamp*.

Return Values

The elapsed time in seconds.

UTL_GetDicomDate

Name

UTL_GetDicomDate -retrieve the current date and convert to DICOM format.

Synopsis

```
void UTL_GetDicomDate( *datestr );
```

datestr A string to hold the converted date.

Description

The current date is retrieved with the time and localtime UNIX functions, and converted into the following string format: "yyyymmdd".

Notes

None.

Return Values

None.

UTL_GetDicomTime

Name

UTL_GetDicomTime -this function retrieves the current DICOM time.

Synopsis

```
void UTL_GetDicomTime( char *timestr )
```

timestr The string which will receive the DICOM time.

Description

The current time is retrieved with the time and localtime unix functions, and converted into the following string format: "hhmmss.fffff".

Notes

None.

Return Values

None.

UTL_GetTimeStamp

Name

UTL_GetTimeStamp - get a time stamp to be used to compute elapsed time

Synopsis

```
void* UTL_GetTimeStamp()
```

Description

UTL_GetTimeStamp determines the current time of day and returns an opaque pointer to the caller. This opaque pointer contains the current time of day and will be used by *UTL_DeltaTime* to compute elapsed time.

Notes

Use *UTL_ReleaseTimeStamp* to release the *timeStamp*.

Return Values

An opaque pointer to a time stamp

UTL_RegexMatch

Name

UTL_RegexMatch -perform a DICOM regular expression match with the specified string.

Synopsis

```
CONDITION UTL_RegexMatch( char *regex, char *stm);
```

<i>regex</i>	The DICOM regular expression to try and match
<i>stm</i>	The input string to match.

Description

A simple function to to perform a DICOM regular expression match with the specified string, stm. The semantics of the DICOM patterns must be altered slightly to work correctly with regex under UNIX.

Notes

None.

Return Values

UTL_MATCH
UTL_NOMATCH

UTL_ReleaseTimeStamp

Name

UTL_ReleaseTimeStamp -release the time stamp structure allocated by *UTL_GetTimeStamp*

Synopsis

```
void UTL_ReleaseTimeStamp( void *timeStamp )
```

timeStamp The timestamp value returned by *UTL_GetTimeStamp*

Description

UTL_ReleaseTimeStamp frees the structure allocated by *UTL_GetTimeStamp*.

Notes

Return Values

None

UTL_SqueezeBlanks

Name

UTL_SqueezeBlanks -squeeze blanks out of a string in place.

Synopsis

```
CONDITION UTL_SqueezeBlanks( char *s);
```

s The string to be squeezed.

Description

Upon return, *s* will contain no blanks.

Notes

None.

Return Values

None.

UTL_TimeMatch

Name

UTL_TimeMatch -match a time range as specified in the DICOM standard.

Synopsis

```
CONDITION UTL_TimeMatch( char *timestring, char *stm );
```

<i>timestring</i>	The range to be matched.
<i>stm</i>	The time to be compared against the above range.

Description

This routine was written specifically to match a DICOM time string (*stm*) against a DICOM time range (*timestring*). This routine handles the following ranges: <time>, <time>-, -<time>, and <time1>-<time2>.

Notes

None.

Return Values

UTL_MATCH
UTL_NOMATCH