**White Paper**

**Date/time manipulation using the**

**DATE and STRDATE functions**

Date revised: <Date: January 09, 2009>

Revised By: <Wayne Wilson>

Licensed Materials - Property of IBM

IBM® Sterling Gentran:Server for Microsoft Windows ®

© Copyright IBM Corp. 1990, 2013. All Rights Reserved.

US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

**TABLE OF CONTENTS**

[Purpose 3](#_Toc489455760)

[Background 3](#_Toc489455761)

[STRDATE 3](#_Toc489455762)

[Syntax: 3](#_Toc489455763)

[DATE: 4](#_Toc489455764)

[Syntax 1: 4](#_Toc489455765)

[Syntax 2: 4](#_Toc489455766)

[Format Specifiers 5](#_Toc489455767)

[Adding or Subtracting from date/time values 6](#_Toc489455768)

[Obtaining the Current Date/Time through Extended rule 6](#_Toc489455769)

# Purpose

To introduce the use of the extended rule syntax for converting Date/Time formatted fields to strings (STRDATE) and string formatted fields to dates and time (DATE). Also covered are the parameters used in the syntax to convert from one format to the other and how to add and subtract from Date/Time formatted fields.

# Background

To give the user a starting point to add extended rule functionality to a map.

# STRDATE

The STRDATE function converts a datetime type into a string using a format that you specify. This function allows you to include static characters such as a slash (/), which gives you access to full date support.

The STRDATE function can be used to:

* Convert a date/time value to a string to update it to a string column in a database table
* Convert a date/time value to a string to display the value in a messagebox

## Syntax:

STRDATE (datetime,"format", string);

where: datetime = datetime variable/field

format = date format

string = string variable/field

*Example 1:*

//Convert a Date field (#Date\_Field) to a string value to populate the Ref Data column.

STRDATE (#Date\_Field,"%y/%m/%d",#String\_Field);

UPDATE DOCUMENT SET REFERENCEDATA = #String\_Field;

*Example 2:*

//Convert a Date field (#Date\_Field) to a string value to populate the Document Name column.

STRDATE (#Date\_Field,"%y/%m/%d",#String\_Field);

UPDATE PROCESSDATA SET XPATHRESULT = #String\_Field WHERE XPATH = “PO\_DATE”;

*Example 3:*

//Convert a Date field (#Date\_Field) to a string value to use in a Messagebox extended rule.

STRING [100] msg;

STRDATE (#Date\_Field,"%y/%m/%d",#String\_Field);

msg = “PO DATE: “ + #String\_Field;

MESSAGEBOX(msg,0);

# DATE:

The DATE format enables you to convert any string format type into a datetime format type by indicating a format mask ("%y/%m/%d") along with the string ("2007/04/06") you want to convert.

The DATE function can be used to:

* Convert a string value into a date/time value (Syntax 1)
* Hardcode a date/time value into a variable or field (Syntax 2)

## Syntax 1:

DATETIME = DATE("format", string);

where: datetime = datetime variable

format = date format

string = string variable

*Example:*

//Hard code the value contained in a string field (#string\_date) into the date variable ‘d’

DATETIME d;

d = DATE(0,0,0); //initialize the date variable

d = DATE("%Y/%m/%d", #string\_date);

## Syntax 2:

DATETIME = DATE(Number);

where: Number = date values as numbers separated by commas.

The order of the numbers will need to match the Format specified in the Validation area of the #Date\_field the hard coded value is to be stored in. The value for Year has to be 4 characters long. The value for the month is the number 1-12, and the day is a number 1-31.

*Example 1:*

//Hard code the date 20070406 into the date variable ‘d’ and store it to #Date\_field //(YYYYMMDD)

DATETIME d;

d = DATE(0,0,0); //initialize the date variable

d = DATE(2007,4,6);

#Date\_field = d;

*Example 2:*

//Hard code the date 20070406 with a time of 12:00 into the date variable ‘d’ and store it to //#Date\_field (YYMMDDHHMM)

DATETIME d;

d = DATE(0,0,0,0,0); //initialize the date variable

d = DATE(2007,4,6,12,0);

#Date\_field = d;

# Format Specifiers

This table lists the format specifiers available for the DATE and STRDATE functions. These specifiers are used with the two functions (DATE and STRDATE) to indicate what format the converted date/time will be converted.

Please check the Help text in the version of Gentran you are using.

|  |  |
| --- | --- |
| Format Specifier | Description |
| %8 | ISO-8601 date format. Valid format is YYYYMMDD'T'HHMMSS'.'sssZ  Note  This date format cannot be combined with any other format specifier. |
| %a | Abbreviated weekday name. |
| %A | Full weekday name. |
| %b | Abbreviated month name. |
| %B | Full month name. |
| %d | Day of the month as a decimal number (01 - 31). |
| %H | Hour in 24-hour format (00 - 23). |
| %I | Hour in 12-hour format (01 - 12). |
| %j | Day of the year as a decimal number (001 - 366). |
| %m | Month as a decimal number (01 - 12). |
| %M | Minute as a decimal number (00 - 59). |
| %S | Second as a decimal number (00 - 59) |
| %U | Week of the year as a decimal number, with Sunday as the first day of the week (00 - 51). |
| %w | Weekday as a decimal number (0 - 6, with Sunday as "0"). |
| %W | Week of the year as a decimal number, with Monday as the first day of the week (00 - 51). |
| %y | Year without the century as a decimal number (00 - 99). |
| %Y | Year with the century as a decimal number. |
| %% | Percent sign. |

# Adding or Subtracting from date/time values

You can use the << operator to modify a date/time value to add or subtract time increments (e.g., days, weeks, years).

Adding

datetime d;

d= #date\_field;

d = d << weeks (2);

//This adds 2 weeks to d.

Subtracting

datetime d;

d= #date\_field;

d = d << weeks (-2);

//This subtracts 2 weeks from d.

The keywords available for adding or subtracting time increments are:

* Years
* Months
* Days
* Hours
* Minutes
* Seconds

# Obtaining the Current Date/Time through Extended rule

For Gentran Server for Windows you will use the GentranEx.DLL

Object obRules;

obRules = CreateObject("GentranEx.Rules");

#Date\_field = obRules.GetDate();

#Time\_field = obRules.GetTime();

deleteobject(obRules);

For Gentran Integration Suite you will use a java command

object java\_date;  
datetime current\_date\_time;

java\_date = new("java.util.Date");

current\_date\_time = java\_date;

#DATE\_FIELD = current\_date\_time;