
Artículo

[Nancy Martínez](#) · 28 jun, 2019 Lectura de 9 min

Administración de la fecha y hora durante las operaciones en Caché

¡Hola Comunidad!

En este artículo encontrarán algunos ejemplos de conversiones y operaciones que les pueden resultar útiles. También incluyo enlaces a la documentación donde se puede obtener más información.

Cuando escribí esto, la hora del Este "Eastern Daylight Time" estaba activa en el Caché de mi sistema.

Cómo guarda Caché la hora y la fecha

Caché tiene un formato simple para la hora, con un rango más amplio para las fechas reconocidas, en comparación con otras tecnologías.

La hora actual se actualiza con la variable especial \$HOROLOG (\$H):

```
USER>WRITE $H
```

```
64146,54027
```

```
USER>
```

El primer número entero corresponde al número de días transcurridos desde el 31 de diciembre de 1840. El segundo número entero corresponde al número de segundos transcurridos desde la medianoche del día en curso.

También se puede obtener la hora y la fecha actuales con \$SYSTEM.SYS.Horolog().

Cómo establecer un registro de la hora

\$HOROLOG registra el tiempo transcurrido en segundos. \$ZTIMESTAMP es un formato similar a \$HOROLOG, pero registra las fracciones de segundo en una porción de tiempo y guarda el Tiempo Universal Coordinado (UTC), en lugar de la hora local. La precisión de las fracciones de segundo depende de la plataforma que se esté usando.

Por esta razón, \$ZTIMESTAMP proporciona un registro de las horas que es uniforme en todas las zonas horarias. El registro de la hora que ve en cualquier momento en el tiempo podría tener una fecha y hora diferente a su hora local actual. En este ejemplo, mi hora local es la hora del Este, que tiene cuatro horas de retraso respecto al Tiempo Universal Coordinado (UTC).

```
WRITE !,"$ZTIMESTAMP: "__ZTIMESTAMP_" $HOROLOG: "__$HOROLOG
```

```
$ZTIMESTAMP: 64183,53760.475 $HOROLOG: 64183,39360
```

La diferencia (sin contar las fracciones de segundo) es de 14400 segundos. Por lo tanto, mi \$HOROLOG tiene un "retraso" de cuatro horas con \$ZTIMESTAMP.

Cómo realizar conversiones desde el formato interno al formato de visualización

Puede utilizar \$ZDATETIME. Realizar conversiones de la fecha y la hora actuales desde el formato interno puede ser tan sencillo como

```
WRITE !, "With default date and time format: ", $ZDATETIME($HOROLOG)
```

```
With default date and time format: 09/22/2016 10:56:00
```

Esto toma la configuración predeterminada de la configuración local de Caché y del soporte de idiomas nativos (NLS).

El segundo y tercer argumento (opcional) tienen la función de especificar el formato de la fecha y la hora.

```
WRITE !, "With dformat 5 and tformat 5: ", $ZDATETIME($HOROLOG,5,5)
```

```
With dformat 5 and tformat 5: Sep 22, 2016T10:56:00-04:00
```

Por ejemplo, Time format 7 muestra la hora en el formato del Tiempo Universal Coordinado, como puede verse aquí.

```
WRITE !, "With dformat 5 and tformat 7: ", $ZDATETIME($HOROLOG,5,7)
```

```
With dformat 5 and tformat 7: Sep 22, 2016T14:56:00Z
```

Además de los formatos para la fecha y la hora, existen muchos otros argumentos opcionales que le permitirán controlar la visualización. Por ejemplo, puede:

- Especificar los límites inferior y superior de las fechas válidas si también están permitidos por la ubicación actual
- Establecer si los años se visualizan con dos o cuatro dígitos
- Controlar la manera en que se visualizarán los errores

Cómo realizar conversiones entre un formato de visualización y el formato interno de Caché

Utilice \$ZDATETIMEH para ir desde el formato de visualización a un formato interno, como \$HOROLOG. La H que se encuentra al final es un recordatorio de que terminará con un formato \$HOROLOG. Pueden utilizarse muchos formatos de entrada diferentes.

```
SET display = "09/19/2016 05:05 PM"
WRITE !, display_" is "_$ZDATETIMEH(display)_" in internal format"
WRITE !, "Suffixes AM, PM, NOON, MIDNIGHT can be used"
SET startdate = "12/31/1840 12:00 MIDNIGHT"
WRITE !, startdate_" is "_$ZDATETIMEH(startdate)_" in internal format"
```

```
09/19/2016 05:05 PM is 64180,61500 in internal format
```

```
Suffixes AM, PM, NOON, MIDNIGHT can be used
```

```
12/31/1840 12:00 MIDNIGHT is 0,0 in internal format
```

Cómo hacer conversiones UTC hacia y desde la hora local en un formato interno

Puede utilizar \$ZDATETIME y \$ZDATETIMEH con un especificador especial (-3) para el formato de la fecha en el segundo argumento.

La mejor manera para convertir la hora UTC a la hora local en el formato interno de Caché, es utilizar la función \$ZDATETIMEH (con datetime, -3). Aquí, el primer argumento contiene la hora UTC en el formato interno.

```
SET utc1 = $ZTIMESTAMP
SET loctime1 = $ZDATETIMEH(utc1, -3)
WRITE !, "$ZTIMESTAMP returns a UTC time in internal format: ", utc1
WRITE !, "$ZDATETIMEH( ts,-3) converts UTC to local time: ", loctime1
WRITE !, "$ZDATETIME converts this to display formats: ", $ZDATETIME(utc1)
WRITE !, "which is "_$ZDATETIME(loctime1)_" in local time"
```

```
$ZTIMESTAMP returns a UTC time in internal format: 64183,53760.475
```

```
$ZDATETIMEH( ts,-3) converts UTC to local time: 64183,39360.475
```

```
$ZDATETIME converts this to display formats: 09/22/2016 14:56:00
```

```
which is 09/22/2016 10:56:00 in local time
```

Si se requiere hacer conversiones desde la hora local hacia el UTC, de nuevo en el formato interno, utilice `$ZDATETIME`(con `datetime`, -3). Aquí, `datetime` contiene la hora que indica tu zona horaria local en el formato interno.

```
SET loctime2 = $HOROLOG
SET utc2 = $ZDATETIME(loctime2, -3)
WRITE !, "$HOROLOG returns a local time in internal format: ", loctime2
WRITE !, "$ZDATETIME(ts, -3) converts this to UTC: ", utc2
WRITE !, "$ZDATETIME converts this to display formats:"
WRITE !, "Local: ", $ZDATETIME(loctime2)
WRITE !, "UTC: ", $ZDATETIME(utc2)
```

```
$HOROLOG returns a local time in internal format: 64183,39360
```

```
$ZDATETIME(ts, -3) converts this to UTC: 64183,53760
```

```
$ZDATETIME converts this to display formats:
```

```
Local: 09/22/2016 10:56:00
```

```
UTC: 09/22/2016 14:56:00
```

Se debe tener en cuenta estos puntos cuando se realiza conversiones entre la hora local y el UTC:

- Las conversiones entre la hora local y el UTC deben utilizar las reglas vigentes de la zona horaria para la fecha y el lugar especificados. Caché depende del sistema operativo para registrar esos cambios en el tiempo. Si el sistema operativo no realiza esto de forma apropiada, las conversiones no serán correctas.
- Las conversiones de fechas y horas que se realicen en el futuro utilizaran las reglas que conserva actualmente el sistema operativo. Sin embargo, la reglas para las conversiones de los años, que se realicen en el futuro, pueden cambiar.

Cómo determinar la zona horaria en el sistema

Se puede obtener el desfase con la zona horaria actual al examinar el valor de `$ZTIMEZONE` o `%SYSTEM.SYS.TimeZone()`. El sistema operativo establece el valor predeterminado.

```
WRITE !, "$ZTIMEZONE is set to "._$ZTIMEZONE
WRITE !, "%SYSTEM.SYS.TimeZone() returns "._$System.SYS.TimeZone()
```

```
$ZTIMEZONE is set to 300
```

```
%SYSTEM.SYS.TimeZone() returns 300
```

No se debe modificar el valor de la función `$ZTIMEZONE`. Si se hace, podría afectar los resultados de `IsDST()`, `$ZDATETIME`, y `$ZDATETIMEH`, entre muchas otras consecuencias. La hora establecida por el proceso no se actualizará correctamente para el horario de verano. Realizar modificaciones en `$ZTIMEZONE` provocará que los cambios no se apliquen de manera uniforme en la zona horaria que utiliza Caché.

A partir de la versión 2016.1, Caché proporciona el método `$System.Process.TimeZone()`, el cual le permite establecer y recuperar la zona horaria para realizar un proceso específico utilizando la variable de entorno `TZ`. El método devuelve -1 si la variable `TZ` no se estableció.

```
WRITE !,$System.Process.TimeZone()  
WRITE !, "Current Time: " _ $ZDT($H)  
WRITE !, "Set Central Time"  
DO $System.Process.TimeZone("CST6CDT")  
WRITE !, "New current time: " _ $ZDT($H)  
WRITE !, "Current Time Zone: " _ $System.Process.TimeZone()
```

```
-1
```

```
Current Time: 10/03/2016 15:46:04
```

```
Set Central Time
```

```
New current time: 10/03/2016 14:46:04
```

```
Current Time Zone: CST6CDT
```

Cómo determinar si el horario de verano está activo

Utilice `$SYSTEM.Util.IsDST()`. En este caso, Caché también depende del sistema operativo para aplicar las reglas correctas y determinar si el horario de verano está activo.

```
SET dst = $System.Util.IsDST()  
IF (dst = 1) {WRITE !, "DST is in effect"}  
ELSEIF (dst = 0) { WRITE !, "DST is not in effect" }  
ELSE { WRITE !, "DST cannot be determined" }
```

Cómo realizar cálculos con las fechas

Dado que el formato interno de Caché mantiene un recuento de los días y de los segundos que hay en cada día,

usted puede calcular las fechas de forma directa. La función \$PIECE le permite quitar los elementos de la fecha y hora del formato interno.

Esta es una rutina corta que utiliza \$ZDATE y \$ZDATEH para determinar cuál fue el último día del año pasado, de modo que pueda calcular el día actual del año en curso. Esta rutina utiliza métodos en la clase %SYS.NLS para establecer el formato que deseemos, obtener las separaciones para la fecha, y configurar nuevamente los valores predeterminados.

```
DATECALC ; Example of date arithmetic.
W !, "Extracting date and time from $H using $PIECE"
W !, "-----"
set curtime = $H
set today = $PIECE(curtime,"",1)
set now = $PIECE(curtime,"",2)
W !, "Curtime: "_curtime_" Today: "_today_" Now: "_now"

W !, "Counting the days of the year"
W !, "-----"
; set to US format
SET rtn = ##class(%SYS.NLS.Format).SetFormatItem("DateFormat",1)
set sep = ##class(%SYS.NLS.Format).GetFormatItem("DateSeparator")
SET lastyear = ($PIECE($ZDATE($H),sep,3) - 1)
SET start = $ZDATEH("12/31/"_lastyear)
W !, "Today is day "_(today - start)_" of the year"
; put back the original date format
SET rtn=##class(%SYS.NLS.Format).SetFormatItem("DateFormat",rtn)
```

Cómo obtener y establecer otras configuraciones para NLS

Utilice la clase %SYS.NLS.Format para configurar cosas como el formato de la fecha, las fechas máximas y mínimas y realizar otros ajustes. La configuración inicial proviene de la ubicación actual y los cambios que realice con esta clase influirán únicamente en el proceso actual.

Fecha y hora en SQL

Caché proporciona varias funciones en SQL para trabajar con las fechas y las horas. Estas funciones también están disponibles en ObjectScript mediante la clase \$System.SQL.

TODATE: Convierte una cadena de caracteres con formato CHAR o VARCHAR2 en una fecha. Esta función está disponible en ObjectScript utilizando \$System.SQL.TODATE(" string ", " format ")

DAYOFYEAR: Devuelve el día del año para expresar un cierto año, el cual puede estar en varios formatos, como un entero para mostrar una fecha desde \$HOROLOG.

DAYNAME: Devuelve el nombre del día que corresponde con una fecha específica.

```
W $ZDT($H)
```

```
10/12/2016 11:39:19
```

```
w $System.SQL.TODATE( "2016-10-12" , "YYYY-MM-DD" )
```

64203

```
W $System.SQL.DAYOFYEAR(+$H)
```

286

```
W $System.SQL.DAYNAME(+$H)
```

Wednesday

Hay mucha información sobre cómo utilizar estas funciones (y muchas más) en la documentación de Caché. Consulte las referencias en la siguiente sección.

Referencias

Enlaces a los documentos de soporte de InterSystems, disponibles "online":

\$HOROLOG: <http://docs.intersystems.com/latest/csp/docbook/DocBook.UI.Page.cls?KEY=RCOSvhorolog>

\$PIECE: <http://docs.intersystems.com/latest/csp/docbook/DocBook.UI.Page.cls?KEY=RCOSfpiece>

Referencia a las funciones de SQL:

<http://docs.intersystems.com/latest/csp/docbook/DocBook.UI.Page.cls?KEY=RSQLFUNCTIONS>

%SYS.NLS.Format:

<http://docs.intersystems.com/latest/csp/documatic/%25CSP.Documatic.cls?PAGE=CLASS&LIBRARY=%25SYS&CLASSNAME=%25SYS.NLS.Format>

%SYSTEM.Process.TimeZone():

<http://docs.intersystems.com/latest/csp/documatic/%25CSP.Documatic.cls?PAGE=CLASS&LIBRARY=%25SYS&CLASSNAME=%25SYSTEM.Process#METHODTimeZone>

%SYSTEM.SQL:

<http://docs.intersystems.com/latest/csp/documatic/%25CSP.Documatic.cls?PAGE=CLASS&LIBRARY=%25SYS&CLASSNAME=%25SYSTEM.SQL>

%SYSTEM.SYS.Horolog:

<http://docs.intersystems.com/latest/csp/documatic/%25CSP.Documatic.cls?PAGE=CLASS&LIBRARY=%25SYS&CLASSNAME=%25SYSTEM.SYS>

%SYSTEM.Util.IsDST():

<http://docs.intersystems.com/latest/csp/documatic/%25CSP.Documatic.cls?PAGE=CLASS&LIBRARY=%25SYS&CLASSNAME=%25SYSTEM.Util#IsDST>

\$ZDATE: <http://docs.intersystems.com/latest/csp/docbook/DocBook.UI.Page.cls?KEY=RCOSfzdate>

\$ZDATEH: <http://docs.intersystems.com/latest/csp/docbook/DocBook.UI.Page.cls?KEY=RCOSfzdateh>

\$ZDATETIME: <http://docs.intersystems.com/latest/csp/docbook/DocBook.UI.Page.cls?KEY=RCOSfzdatetime>

\$ZDATETIMEH: <http://docs.intersystems.com/latest/csp/docbook/DocBook.UI.Page.cls?KEY=RCOSfzdatetimeh>

[#ObjectScript](#) [#Administración del sistema](#) [#Caché](#)

URL de
fuente: <https://es.community.intersystems.com/post/administraci%C3%B3n-de-la-fecha-y-hora-durante-las-operaciones-en-cach%C3%A9>