


文章  
姚鑫 · 二月 20, 2021  阅读大约需分钟

## 第四十二章 Caché 变量大全 \$ZTIMESTAMP 变量

### 第四十二章 Caché 变量大全 \$ZTIMESTAMP 变量

包含协调世界时间格式的当前日期和时间。

## 大纲

\$ZTIMESTAMP  
\$ZTS

## 描述

\$ZTIMESTAMP包含协调的通用时间值形式的当前日期和时间。这是世界范围内的时间和日期标准；此值很可能与当地的时间(和日期)值不同。

\$ZTIMESTAMP将日期和时间表示为以下格式的字符串：

```
dddddd,sssss.fff
```

其中dddddd是一个整数，指定自1840年12月31日起的天数；sssss是一个整数，指定自当天午夜以来的秒数，fff是一个可变的数字，指定小数秒。这种格式类似于\$HOROLOG，只是\$HOROLOG不包含分数秒。

假设当前日期和时间(世界协调时)如：

```
2018-02-22 15:17:27.984
```

当时，\$ZTIMESTAMP的值为：

```
64701,55047.984
```

\$ZTIMESTAMP报告协调世界时(UTC)，它位于时区。因此，\$ZTIMESTAMP提供了一个跨时区的统一时间戳。这可能不同于本地时间值和本地日期值。

\$ZTIMESTAMP时间值是一个十进制数值，以秒及其分数为单位计算时间。分数秒的位数可能从零到九不等，具体取决于计算机时钟的精度。在视窗系统上，小数精度是三位小数；在UNIX系统上，它是六位十进制数字。\$ZTIMESTAMP在此小数部分中抑制尾随零或尾随小数点。请注意，在午夜后的第一秒内，秒表示为0.fff(例如，0.123)；这个数字不是ObjectScript规范形式(例如，.123)，这会影响这些值的字符串排序顺序。在执行排序操作之前，您可以添加一个加号(+)来强制将数字转换为规范形式。

比较了返回当前日期和时间的多种方法，如所示：

- \$ZTIMESTAMP包含以系统间IRIS存储(\$HOROLOG)格式表示的UTC日期和时间(小数秒)。小数秒以三位精度(在Windows系统上)或六位精度(在UNIX®系统上)表示。
- \$NOW返回当前进程的本地日期和时间；不应用本地时间变(如夏令时)。不带参数值的\$NOW根据\$ZTIMEZONE特殊变量的值确定当地时区。带有参数值的\$NOW返回与指定时区参数对应的时间和日期。\$NOW(0)返回UTC日期和时间。忽略\$ZTIMEZONE的值。\$now返回InterSystems IRIS存储(\$HOROLOG)格式的日期和时间。它包括小数秒；小数位数是当前操作系统支持的最大精度。因此，\$NOW(0)返回的UTC时间可能比\$ZTIMESTAMP返回的秒精度高
- \$HOROLOG包含采用InterSystems IRIS存储格式的本地变量调整日期和时间。它不记录小数秒。\$HOROLOG如解析小数秒取决于操作系统平台：在Windows上，它将任何小数秒四舍五入到一整秒。在UNIX®上，它会截断小数部分。

注意：比较当地时间和UTC时间时要谨慎：

将UTC时间转换为本地时间首选方法是使用\$ZDATETIMEH(UTC, -3)函数。此函数根据当地时间变量进行调整。

- 不能通过简单地添加或减去\$ZTIMEZONE\*60的值来转换本地时间和UTC时间。这是因为，在许多情况，当地时间会根据当地时间的变化进行调整(例如夏令时，它会将当地时间季节性调整一小时)。这些本地时间变量不会反映在\$ZTIMEZONE中。

- UTC时间是使用格林威治子午线上的时区计数来计算的。这和格林威治当地时间不一样。术语格林威治标准时间(GMT)可能会令人混淆；格林威治当地时间在冬季与UTC相同；在夏季，它与UTC相差一个小时。这是因为采用了当地时间变量，即英国夏令时(British Summer Time)。

- 时区与UTC和本地时间的偏差(例如季节转换为夏令时)都会影响日期和时间。从本地时间转换为UTC时间(反之亦然)可能会更改日期和时间。

不能使用SET命令修改特殊变量。尝试这样做会导致<SYNTAX>错误。

## 协调世界时转换

可以使用带有tFormat值7或8的\$ZDATETIME和\$ZDATETIMEH函数将本地时间信息表示为协调世界时(UTC)，如例所示：

```
/// d ##class(PHA.TEST.SpecialVariables).ZTIMESTAMP()  
ClassMethod ZTIMESTAMP()  
{  
    WRITE !,$ZDATETIME($ZTIMESTAMP,1,1,2)  
    WRITE !,$ZDATETIME($HOROLOG,1,7,2)  
    WRITE !,$ZDATETIME($HOROLOG,1,8,2)  
    WRITE !,$ZDATETIME($NOW(),1,7,2)  
    WRITE !,$ZDATETIME($NOW(),1,8,2)  
}
```

```
DHC-APP>d ##class(PHA.TEST.SpecialVariables).ZTIMESTAMP()
```

```
02/10/2021 09:46:32.53  
02/10/2021T09:46:32.00Z  
02/10/2021T09:46Z  
02/10/2021T09:46:32.53Z  
02/10/2021T09:46Z
```

上面的\$ZDATETIME函数都以协调世界时(而不是本地时间)的形式返回当前时间。这些从本地时间转换的时间值可能不同，因为\$Now不会针对本地时间变量进行调整；\$ZTIMESTAMP和\$HOROLOG会针对本地时间变量进行调整，

并可能在必要时相应地调整日期。\$ZTIMESTAMP显示值与tFormat 7或8转换后的显示值不同。Tformat值7和8在时间值之前插入字母“T”，在时间值之后插入字母“Z”。此外，因为\$HOROLOG TIME不包含小数秒，所以上例中精度为2的小数位用零填充。

通过使用以下语法形式之一调用Timestamp()类方法，可以获得与\$ZTIMESTAMP相同的时间戳信息：

```
DHC-APP> WRITE !,$SYSTEM.SYS.TimeStamp()

65785,35395.629
DHC-APP> WRITE !,##class(%SYSTEM.SYS).TimeStamp()

65785,35408.245
```

## 示例

下面的示例将\$ZTIMESTAMP的值转换为本地时间，并将其与本地时间的两种表示形式进行比较：\$NOW()和\$HOROLOG：

```
/// d ##class(PHA.TEST.SpecialVariables).ZTIMESTAMP1()
ClassMethod ZTIMESTAMP1()
{
    SET stamp=$ZTIMESTAMP,clock=$HOROLOG,miliclock=$NOW()
    WRITE !,"????????: ",$ZDATETIME(clock,1,1,2)
    WRITE !,"????????: ",$ZDATETIME(miliclock,1,1,2)
    WRITE !,"UTC?????:   ",$ZDATETIME(stamp,1,1,2)
    IF $PIECE(stamp,"",2) = $PIECE(clock,"",2) {
        WRITE !,"?????UTC??" }
    ELSEIF $PIECE(stamp,"") '= $PIECE(clock,"") {
        WRITE !,"??????" }
    ELSE {
        SET localutc=$ZDATETIMEH(stamp,-3)
        WRITE !,"UTC?????: ",$ZDATETIME(localutc,1,1,2)
    }
    QUIT
}
```

```
DHC-APP>d ##class(PHA.TEST.SpecialVariables).ZTIMESTAMP1()

????????: 02/10/2021 17:54:46.00
????????: 02/10/2021 17:54:46.93
UTC?????: 02/10/2021 09:54:46.93
UTC?????: 02/10/2021 17:54:46.93
```

下面的示例比较了\$ZTIMESTAMP和\$HOROLOG返回的值，并显示了如何将\$ZTIMESTAMP的时间部分。（请注意，在此简单示例中，只针对本地时间变量（如夏令时）进行了一次调整。其他类型局部变量可能会导致时钟秒和戳秒包含不可调和的值。）

```
/// d ##class(PHA.TEST.SpecialVariables).ZTIMESTAMP2()
ClassMethod ZTIMESTAMP2()
{
    SET stamp=$ZTIMESTAMP,clock=$HOROLOG
    WRITE !,"????????: ",$ZDATETIME(clock,1,1,2)
```

```
WRITE !,"UTC?????:    ",$ZDATETIME(stamp,1,1,2)
IF $PIECE(stamp,",") '= $PIECE(clock,",") {
  WRITE !,"?????" }
  SET clocksecs=$EXTRACT(clock,7,11)
  SET stampsecs=$EXTRACT(stamp,7,11)-($ZTIMEZONE*60)
  IF clocksecs=stampsecs {
    WRITE !,"???????"
    WRITE !,"?????????" }
  ELSE {
    SET stampsecs=stampsecs+3600
  }
  IF clocksecs=stampsecs {
    WRITE !,"?????:"
    WRITE !,"????????????1??" }
  ELSE {
    WRITE !,"?????????????"
  }
}
QUIT
}
```

```
DHC-APP>d ##class(PHA.TEST.SpecialVariables).ZTIMESTAMP2()
```

```
????????: 02/10/2021 17:58:16.00
UTC?????: 02/10/2021 09:58:16.85
?????????
??????????
```

[#Caché #InterSystems IRIS #InterSystems IRIS for Health](#)

源 URL: <https://cn.community.intersystems.com/post/%E7%AC%AC%E5%9B%9B%E5%8D%81%E4%BA%8C%E7%AB%A0-cach%C3%A9-%E5%8F%98%E9%87%8F%E5%A4%A7%E5%85%A8-ztimestamp-%E5%8F%98%E9%87%8F>