文章
姚 鑫 · 二月 21, 2021 阅读大约需 7 分钟

# 第四十三章 Caché 变量大全 $ZTIMEZONE 变量

# 第四十三章 Caché 变量大全 $ZTIMEZONE 变量

包含格林威治子午线的时区偏移量。

## 大纲

```
$ZTIMEZONE
$ZTZ
```

## 描述

$ZTIMEZONE可以通过两种方式使用：

- 返回计算机的本地时区偏移量。
- 为当前进程设置本地时区偏移量。

$ZTIMEZONE包含从格林威治子午线偏移的时区（以分钟为单位）。 （格林威治子午线包括整个英国和爱尔兰。）此偏移量表示为-1440到1440范围内的有符号整数。格林威治以西的时区指定为正数；格林威治东部的时区指定为负数。 （时区必须以分钟为单位，因为并非所有时区都以小时为单位。）默认情况下，$ZTIMEZONE初始化为计算机操作系统设置的时区。

注意：$ZTIMEZONE将本地时间调整为固定的偏移量。它不适应夏令时或其他当地时间的变化。 InterSystems IRIS从基础操作系统获取本地时间，该操作系统将本地时间变体应用于为该计算机配置的位置。因此，使用$ZTIMEZONE调整的本地时间将从配置的语言环境中获取其本地时间变化，而不是在$ZTIMEZONE中指定的时区。

使用格林威治子午线（$ZTIMEZONE = 0）的时区计数来计算UTC时间。它与当地格林威治时间不同。格林威治标准时间（GMT）一词可能令人困惑；格林威治的当地时间与冬季的UTC相同。在夏季，它与UTC的差异为一小时。这是因为应用了称为英国夏令时的本地时间变体。

对于使用$ZTIMEZONE的函数和程序，经过的本地时间始终是连续的，但是时间值可能需要季节性调整以与本地时钟时间相对应。

## 设定时区

可以使用$ZTIMEZONE设置当前InterSystems
IRIS进程使用的时区。设置$ZTIMEZONE不会更改默认的InterSystems IRIS时区或计算机的时区设置。

注意：更改$ZTIMEZONE特殊变量是为某些特殊情况设计的功能。更改$ZTIMEZONE并不是更改InterSystems IRIS用于本地日期/时间操作的时区的一致方法。除非已准备好处理所有导致的不一致的程序，否则不应更改$ZTIMEZONE特殊变量。

在某些平台上，更改时区可能比更改$ZTIMEZONE特殊变量更好。如果平台具有特定于进程的时区设置（例如POSIX系统上的TZ环境变量），则进行外部系统调用来更改特定于进程的时区可能比更改$ZTIMEZONE更好。在操作系

统级别更改特定于流程的时区将更改UTC的本地时间偏移，并应用确定何时应用本地时变的相应算法。如果默认系统时区在北半球，而所需的过程时区在南半球，则这尤其重要。更改$ZTIMEZONE会将本地时间更改为与UTC偏移的新时区，但是确定何时应用本地时变的算法保持不变。

使用SET命令将$ZTIMEZONE设置为指定的带符号整数分钟数。数字的前导零和小数部分将被忽略。如果在设置$ZTIMEZONE时指定非数字值或无值，则InterSystems IRIS会将$ZTIMEZONE设置为0（格林威治子午线）。

例如，北美东部标准时间（EST）在格林威治以西五个小时。因此，要将当前的InterSystems IRIS流程设置为EST，则需要指定300分钟。要指定格林威治以东一小时的时区，请指定-60分钟。要指定格林威治本身，可以指定0分钟。

设置$ZTIMEZONE：

- 影响无参数的$NOW（）当地时间值。它更改了$NOW（）的时间部分，并且此时间更改也可以更改当前进程的$NOW（）的日期部分。 $NOW（）精确地反映了$ZTIMEZONE设置，其值未针对本地时变进行调整。
- 影响$HOROLOG当地时间值。 $HOROLOG从$ZTIMEZONE获取其时区值，然后季节性调整本地时间，例如夏令时。因此，$HOROLOG始终符合本地时钟时间，但全年的$HOROLOG经过时间不是连续的。
- 影响%SYSTEM.Util类方法IsDST（），UTCtoLocalWithZTIMEZONE（）和LocalWithZTIMEZONEtoUTC（）。
- 不会影响$ZTIMESTAMP或$ZHOROLOG值。
- 不会影响$ZDATE，$ZDATEH，$ZDATETIME，$ZDATETIMEH，$ZTIME和$ZTIMEH函数执行的日期和时间格式转换。
- 不会影响$NOW（n）函数。
- 不会影响%SYSTEM.Process类的FixedDate（）类方法，该方法将$HOROLOG中的日期设置为固定值。

更改$ZTIMEZONE后发生以下异常：
- $ZDATETIME（$HOROLOG，1,7）通常返回UTC时间，但是如果$ZTIMEZONE已更改，它将不返回UTC时间。
- 如果$ZTIMEZONE已更改，$ZDATETIME（$HOROLOG，1,5）将不会返回正确的时区偏移量。
- 如果$ZTIMEZONE已更改，则本地时间和UTC时间之间的$ZDATETIME（$HOROLOG，-3）和$ZDATETIMEH（$ZTIMESTAMP，-3）转换将不正确。

## 其他时区方法

可以通过调用TimeZone（）类方法来获取相同的时区信息，如下所示：

```
DHC-APP>WRITE $SYSTEM.SYS.TimeZone()
-480
```

可以使用tformat值为5或6的$ZDATETIME和$ZDATETIMEH函数，将本地时间变化作为日期和时间字符串的一部分返回，如以下示例所示

```
DHC-APP>WRITE !,$ZDATETIME($HOROLOG,1,5)

02/10/2021T18:24:21+08:00
```

该字符串的最后一部分（+08:00）表示系统的本地时间变化设置，以格林威治子午线为单位，以小时和分钟为单位进行偏移。注意，这种变化不一定是时区偏移量。在上述情况下，时区位于格林威治（-5：00）西部5小时，但是本地时区（夏令时）将时区时间偏移一小时到-04：00。设置$ZTIMEZONE将更改$ZDATETIME（$HOROLOG，1,5）返回的当前处理日期和时间，但不会更改系统本地时间变化设置。

## $ZDATETIMEH使用时区设置

可以将$ZDATETIMEH与dformat = -3一起使用，以将协调世界时（UTC）日期和时间值转换为本地时间。该函数将

UTC值（$ZTIMESTAMP）作为输入。它使用本地时区设置来返回相应的日期和时间，并在适用的情况下应用本地时变（例如夏时制）。

```
/// d ##class(PHA.TEST.SpecialVariables).ZTIMEZONE()
ClassMethod ZTIMEZONE()
{
    SET clock=$HOROLOG
    SET stamp=$ZDATETIMEH($ZTIMESTAMP,-3)
    WRITE !,"??/???????: ",$ZDATETIME(clock,1,1,2)
    WRITE !,"UTC/???????:   ",$ZDATETIME(stamp,1,1,2)
}
```

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

??/???????: 02/10/2021 18:31:27.00
UTC/???????:   02/10/2021 18:31:27.94
```

## 使用$ZTIMEZONE的本地/UTC转换方法

%SYSTEM.Util类的两个类方法在本地日期和时间与UTC日期和时间之间进行转换：UTCtoLocalWithZTIMEZONE()和LocalWithZTIMEZONEtoUTC()。这些方法受$ZTIMEZONE更改的影响。

```
/// d ##class(PHA.TEST.SpecialVariables).ZTIMEZONE1()
ClassMethod ZTIMEZONE1()
{
    WRITE $SYSTEM.Util.UTCtoLocalWithZTIMEZONE($ZTIMESTAMP),!
    WRITE $HOROLOG,!
    WRITE $SYSTEM.Util.LocalWithZTIMEZONEtoUTC($H),!
    WRITE $ZTIMESTAMP
}
```

```
DHC-APP>d ##class(PHA.TEST.SpecialVariables).ZTIMEZONE1()
65785,66819.613
65785,66819
65785,38019
65785,38019.614
```

# 示例

以下示例返回当前时区：

```
/// d ##class(PHA.TEST.SpecialVariables).ZTIMEZONE2()
ClassMethod ZTIMEZONE2()
{
    SET zone=$ZTIMEZONE
    IF zone=0 {
        WRITE !,"???????????"
    } ELSEIF zone>0 {
        WRITE !,"??? ",zone/60," ??????"
    } ELSE {
        WRITE !,"??? ",(-zone)/60," ??????"
```

```
    }
}


DHC-APP>d ##class(PHA.TEST.SpecialVariables).ZTIMEZONE2()

??? 8 ??????
```

以下示例显示了设置时区可以更改日期和时间：

```
/// d ##class(PHA.TEST.SpecialVariables).ZTIMEZONE3()
ClassMethod ZTIMEZONE3()
{
    SET zonesave=$ZTIMEZONE
    WRITE !,"???????:  ",$ZDATE($HOROLOG)
    IF $ZTIMEZONE=0 {
        SET $ZTIMEZONE=720
    } ELSEIF $ZTIMEZONE>0 {
        SET $ZTIMEZONE=($ZTIMEZONE-720)
    } ELSE {
        SET $ZTIMEZONE=($ZTIMEZONE+720)
    }
    WRITE !,"Date halfway around the world: ",$ZDATE($HOROLOG)
    WRITE !,"??????????: ",$ZDATE($ZTIMESTAMP)
    SET $ZTIMEZONE=zonesave
}



DHC-APP> d ##class(PHA.TEST.SpecialVariables).ZTIMEZONE3()

???????:  02/10/2021
Date halfway around the world: 02/10/2021
??????????: 02/10/2021
```

以下示例确定本地时间是否与时区时间相同:

```
/// d ##class(PHA.TEST.SpecialVariables).ZTIMEZONE4()
ClassMethod ZTIMEZONE4()
{
    SET localnow=$HOROLOG, stamp=$ZTIMESTAMP
    WRITE !,"???????: ",$ZDATETIME(localnow,1,1)
    SET clocksecs=$PIECE(localnow,",",2)
    SET stampsecs=$EXTRACT(stamp,7,11)-($ZTIMEZONE*60)
    IF clocksecs=stampsecs {
        WRITE !,"????????:"
        WRITE !,"?????????"
    } ELSE {
        IF clocksecs=stampsecs+3600 {
            WRITE !,"?????:"
            WRITE !,"???????1???????"
        } ELSE {
            WRITE !,"??????????"
            WRITE !,(clocksecs-stampsecs)/60," ????"
        }
```

```
    }
    QUIT
}


DHC-APP>d ##class(PHA.TEST.SpecialVariables).ZTIMEZONE4()

???????: 02/10/2021 18:40:21
????????:
?????????
```

#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%B8%89%E7%AB%A0-cach%C3%A9-%E5%8F%98%E9%87%8F%E5%A4%A7%E5%85%A8-ztimezone-%E5%8F%98%E9%87%8F