文章
姚 鑫 · 二月 22, 2021 阅读大约需 7 分钟

第四十四章 Caché 变量大全 $ZTRAP 变量

第四十四章 Caché 变量大全 $ZTRAP 变量

包含当前错误陷阱处理程序的名称。

大纲

$ZTRAP
$ZT

描述

$ZTRAP包含当前错误陷阱处理程序的行标签名和/或例程名。有三种方法可以设置$ZTRAP:

- SET $ZTRAP= “ location ”
- SET $ZTRAP= “ *location ”
- SET $ZTRAP= “ ^%ET ” or “ ^%ETN ”

在这里，位置可以指定为标签(当前例程中的行标签)、^routine(指定外部例程的开始)或label^routine(指定外部例程中的指定标签)。

然而，$ZTRAP=label^routine不能用于程序块。过程块中的$ZTRAP不能用于转到过程体之外的位置；过程块中的$ZTRAP只能引用该过程块中的一个位置。

Location

使用设置命令，可以将位置指定为带引号的字符串。

- 在例程中，可以将位置指定为标签(当前例程中的行标签)、^routine(指定外部例程的开始)或label^routine(指定外部例程中的指定标签)。不要在引用过程或过程中的标签的例程中指定位置。这是一个无效位置；当InterSystems IRIS试图执行$ZTRAP时，会导致运行时错误。
- 在过程中，可以将位置指定为标签；过程块中私有标签。过程块中的$ZTRAP不能用于转到过程体之外的位置；过程块中的$ZTRAP只能引用该过程块中的一个位置。因此，在过程中，不能将$ZTRAP设置为^routine或label^routine.尝试这样做将导致<SYNTAX>错误。

在过程中，将$ZTRAP设置为私有标签名，但是$ZTRAP值不是私有标签名；它是从过程标签(过程的顶部)到私有标签的行位置的偏移量。例如，+17^myproc.

注意：$ZTRAP在某些情况下（而不是在过程中）为label + offset提供传统支持。这个可选的+ offset是一个整数，指定要从label偏移的行数。标签必须在相同的例程中。不建议使用+offset，它可能会导致编译警告错误。InterSystems建议您在指定位置时避免使用行偏移量。

调用过程或IRIS SYS%例程时，不能指定+偏移量。如果尝试这样做，则InterSystems IRIS会发出错误。

$ZTRAP位置必须在当前名称空间中。 $ZTRAP不支持扩展的例程引用。

如果指定了不存在的行标签（当前例程中不存在的位置），则会发生以下情况：
- 显示$ZTRAP：在例程中，$ZTRAP包含label ^
routine。例如，DummyLabel^MyRou。在一个过程中，$TRAP包含最大可能的偏移量：+ 34463 ^ MyProc。
- 调用$ZTRAP：InterSystems IRIS发出<NOLINE>错误消息。

每个堆栈级别可以有其自己的$ZTRAP值。设置$ZTRAP时，系统会将$ZTRAP的值保存为先前的堆栈级别。当前堆栈级别结束时，InterSystems IRIS会恢复该值。要在当前堆栈级别启用错误陷阱，请通过指定$ZTRAP的位置将其设置为错误陷阱处理程序。例如：

```
/// d ##class(PHA.TEST.SpecialVariables).ZTRAP()
ClassMethod ZTRAP()
{
    IF $ZTRAP="" {
        WRITE !,"$ZTRAP not set"
    } ELSE {
        WRITE !,"$ZTRAP already set: ",$ZTRAP
        SET oldtrap=$ZTRAP
    }
    SET $ZTRAP="Etrap1^Handler"
    WRITE !,"$ZTRAP set to: ",$ZTRAP
    //  program code
    SET $ZTRAP=oldtrap
    WRITE !,"$ZTRAP restored to: ",$ZTRAP
}
```

发生错误时，此格式将展开调用堆栈，并将控制权转移到指定的错误陷阱处理程序。

在SqlComputeCode中，不要设置$ZTRAP = $ZTRAP。这可能导致事务处理和错误报告方面的重大问题。

要禁用错误捕获，请将$ZTRAP设置为空字符串（“ ”）。这将清除在当前DO堆栈级别设置的所有错误陷阱。

注意：在“ 终端 ”提示符下使用$ZTRAP仅限于当前代码行。 SET
$ZTRAP命令和生成错误的命令必须在同一行代码中。终端在每个命令行的开头将$ZTRAP还原为系统默认值。

*Location

在例程中，可以选择在发生错误后保留调用堆栈。为此，请在位置之前和双引号内放置一个星号（*）。该表格不适用于程序。尝试这样做会导致<SYNTAX> 错误。只能在不是过程的子例程中使用此示例中的：

```
/// d ##class(PHA.TEST.SpecialVariables).ZTRAP()
ClassMethod ZTRAP()
{
Main
    SET $ZTRAP="*OnError"
    WRITE !,"$ZTRAP set to: ",$ZTRAP
    // program code
OnError
    // Error handling code
    QUIT
}
```

这种格式只会导致转到$ZTRAP中指定的行标签；$STACK和$ESTACK保持不变。$ZTRAP错误处理例程的上下文框架与发生错误的上下文框架相同。但是，InterSystems IRIS会将$ROLES重置为设置$ZTRAP的执行级别的有效值；这会阻止$ZTRAP错误处理程序使用在建立错误处理程序后授予例程的提升权限。完成$ZTRAP错误处理例程后，InterSystems IRIS将堆栈展开到上一个上下文级。这种形式的$ZTRAP对于分析意外错误特别有用。

请注意，星号设置$ZTRAP选项；它不是位置的一部分。因此，在$ZTRAP上执行WRITE或ZZDUMP时不会显示此星号。

^%ETN

在例程中，set $ZTRAP=“ ^%ETN ”将系统提供的错误例程%ETN建立为当前错误捕获处理程序。%ETN在调用它的发生错误的上下文中执行。(%et是%etn的旧名称。它们的功能相同，但%ETN的效率略高一些。)。^%ETN错误处理程序的行为总是前缀星号(*)。

因为过程块中的$ZTRAP不能用于转到过程主体之外的位置，所以不能在过程中使用SET $ZTRAP=“ ^%ETN ”。尝试这样做会导致<SYNTAX>错误。

TRY / CATCH 与 $ZTRAP

不能在TRY块内设置$ZTRAP。尝试这样做会生成编译错误。可以在TRY块之前或在CATCH块内设置$ZTRAP。

无论之前是否设置了$ZTRAP，TRY块中发生的错误都由CATCH块处理。CATCH块内发生的错误由当前错误捕获处理程序处理。

下面的第一个示例显示了TRY块中发生的错误。下面的第二个示例显示了try块中引发的异常。在这两种情况下，都会采用CATCH块，而不是$ZTRAP：

```
/// d ##class(PHA.TEST.SpecialVariables).ZTRAP()
ClassMethod ZTRAP()
{
    SET $ZTRAP="Ztrap"
    TRY { WRITE 1/0 }    /* divide-by-zero error */
    CATCH { WRITE "Catch taken" }
    QUIT
Ztrap
    WRITE "$ZTRAP taken"
    SET $ZTRAP=""
    QUIT
}
```

```
DHC-APP>d ##class(PHA.TEST.SpecialVariables).ZTRAP()
Catch taken
```

```
/// d ##class(PHA.TEST.SpecialVariables).ZTRAP1()
ClassMethod ZTRAP1()
{
    SET $ZTRAP="Ztrap"
    TRY {
        SET myvar=##class(Sample.MyException).%New("Example Error",999,,errdatazero)
        WRITE !,"Throwing an exception!",!
        THROW myvar
        QUIT
    } CATCH {
        WRITE "Catch taken"
```

```
    }
    QUIT
Ztrap
    WRITE "$ZTRAP taken"
    SET $ZTRAP=""
    QUIT
}
```

```
DHC-APP>d ##class(PHA.TEST.SpecialVariables).ZTRAP1()
Catch taken
```

但是，try块可以调用设置和使用$ZTRAP的代码。在下面的示例中，$ZTRAP而不是CATCH块捕获被零除错误：

```
/// d ##class(PHA.TEST.SpecialVariables).ZTRAP2()
ClassMethod ZTRAP2()
{
    TRY { DO Errsub }
    CATCH { WRITE "Catch taken" }
    QUIT
Errsub
    SET $ZTRAP="Ztrap"
    WRITE 1/0   /* divide-by-zero error */
    QUIT
Ztrap
    WRITE "$ZTRAP taken"
    SET $ZTRAP=""
    QUIT
}
```

```
DHC-APP>d ##class(PHA.TEST.SpecialVariables).ZTRAP2()
$ZTRAP taken
```

CATCH块中的Throw命令还可以调用$ZTRAP错误处理程序。

示例

下面的示例将$ZTRAP设置为此程序中的OnError例程。然后，它调用发生错误的Suba(尝试将数字除以0)。当错误发生时，InterSystems IRIS调用$ZTRAP中指定的OnError例程。OnError在设置$ZTRAP的上下文级别调用。因为OnError与Main处于相同的上下文级别，所以执行不会返回Main。

```
/// d ##class(PHA.TEST.SpecialVariables).ZTRAP3()
ClassMethod ZTRAP3()
{
Main
    NEW $ESTACK
    SET $ZTRAP="OnError"
    WRITE !,"$ZTRAP set to: ",$ZTRAP
    WRITE !,"Main $ESTACK= ",$ESTACK   // 0
    WRITE !,"Main $ECODE= ",$ECODE
    DO SubA
```

```
    WRITE !,"Returned from SubA"   // not executed
    WRITE !,"MainReturn $ECODE= ",$ECODE
    QUIT
SubA
    WRITE !,"SubA $ESTACK= ",$ESTACK   // 1
    WRITE !,6/0    // Error: division by zero
    WRITE !,"fine with me"
    QUIT
OnError
    WRITE !,"OnError $ESTACK= ",$ESTACK   // 0
    WRITE !,"$ECODE= ",$ECODE
    QUIT
}


DHC-APP>d ##class(PHA.TEST.SpecialVariables).ZTRAP3()

$ZTRAP set to: +970^PHA.TEST.SpecialVariables.1
Main $ESTACK= 0
Main $ECODE= ,ZSYNTAX,ZSYNTAX,ZSYNTAX,ZMETHOD DOES NOT EXIST,M9,M6,M9,
SubA $ESTACK= 1

OnError $ESTACK= 0
$ECODE= ,ZSYNTAX,ZSYNTAX,ZSYNTAX,ZMETHOD DOES NOT EXIST,M9,M6,M9,M9,
```

下面的示例与前面的示例相同，但有一个例外：$ZTRAP位置前面有一个星号(*)。当错误发生在SUBA中时，此星号会导致InterSystems IRIS在SUBA(发生错误的地方)的上下文级调用OnError例程，而不是在Main(设置$ZTRAP的地方)的上下文级调用OnError例程。因此，当OnError完成时，执行将在do命令之后的行返回到Main。

```
/// d ##class(PHA.TEST.SpecialVariables).ZTRAP4()
ClassMethod ZTRAP4()
{
Main
    NEW $ESTACK
    SET $ZTRAP="*OnError"
    WRITE !,"$ZTRAP set to: ",$ZTRAP
    WRITE !,"Main $ESTACK= ",$ESTACK   // 0
    WRITE !,"Main $ECODE= ",$ECODE
    DO SubA
    WRITE !,"Returned from SubA"   // executed
    WRITE !,"MainReturn $ECODE= ",$ECODE
    QUIT
SubA
    WRITE !,"SubA $ESTACK= ",$ESTACK   // 1
    WRITE !,6/0    // Error: division by zero
    WRITE !,"fine with me"
    QUIT
OnError
    WRITE !,"OnError $ESTACK= ",$ESTACK   // 1
    WRITE !,"$ECODE= ",$ECODE
    QUIT
}
```

#Caché #InterSystems IRIS #InterSystems IRIS for Health

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