

### 文章

姚鑫 · 二月 24, 2021 阅读大约需 6 分钟

## 第四十六章 Caché 变量大全 ^\$GLOBAL 变量

## 第四十六章 Caché 变量大全 ^\$GLOBAL 变量

提供有关全局变量和进程私有全局变量的信息。

### 大纲

`^$|nspace|GLOBAL(global_name)`

`^$|nspace|G(global_name)`

`^$||GLOBAL(global_name)`

`^$||G(global_name)`

### 参数

- `|nspace|` 或 `[nspace]` - 可选-扩展SSVN引用，可以是显式名称空间名称，也可以是隐含名称空间。必须计算为带引号的字符串，该字符串括在方括号(`[ " nspace " ]`)或竖线(`| " nspace " |`)中。命名空间名称不区分大小写；它们以大写字母存储和显示。
- `globalname` 计算结果为包含无下标全局名称的字符串的表达式。全局名称区分大小写。使用`^$||global()`语法时，与进程专用全局名称相对应的无下标全局名称：`^a`表示`^|a`。

### 描述

可以将`^$GLOBAL`用作`$DATA`、`$ORDER`和`$QUERY`函数的参数，以返回有关当前名称空间(默认名称空间)或指定名称空间中是否存在全局变量的信息。还可以使用`^$global`返回有关存在进程私有全局变量的信息。

### 进程私有全局变量

可以使用`^$global`获取有关所有命名空间中是否存在进程私有全局变量的信息。可以将进程专用全局的查找指定为`^$|global`或`^$| " ^ " |global`。

例如，要获取有关进程私有全局`^|a`及其后代的信息，可以指定`$DATA(^$||global( " ^a " ))`。进程私有全局变量不是特定于名称空间的，因此在定义进程私有全局变量时，无论当前名称空间如何，此查找都会返回有关`^|a`的信息。

请注意，`^$GLOBAL`不支持在`GLOBALNAME`本身中指定进程专用全局语法。使用进程专用全局语法指定`GLOBALNAME`会导致`<NAME>`错误。

### 参数

`nspace`

此可选参数允许`^$GLOBAL`查找在另一个命名空间中定义的`GLOBALNAME`。这称为扩展SSVN参考。可以显式地将命名空间名称指定为带引号的字符串文字、变量，也可以通过指定隐含的命名空间来指定。命名空间名称不区分大

小写。可以使用方括号语法[ “ user ” ]或环境语法| “ user ” |。Nspace分隔符前后不允许有空格

可以使用以下方法测试是否定义了命名空间：

```
DHC-APP>WRITE ##class(%SYS.Namespace).Exists("USER")
1
DHC-APP>WRITE ##class(%SYS.Namespace).Exists("LOSER")
0
```

以使用\$NAMESPACE特殊变量来确定当前名称空间。更改当前名称空间的首选方式是新建\$NAMESPACE，然后设置\$NAMESPACE= “ nspace ename ”。

### globalname

计算结果为包含无下标全局名称的字符串的表达式。全局变量区分大小写。

- ^\$global( “ ^a ” ) : globalname “ ^a ” 在当前名称空间中查找此全局名称及其后代。它不查找进程私有全局 “ ^||a ”。

- ^\$|“USER”|GLOBAL(“^a”) : globalname “^a”在 “ user ” 名称空间中查找此全局名称及其后代。它不查找进程私有全局“^||a”。

- ^\$||GLOBAL(“^a”) : globalname “^a”在所有名称空间中查找进程私有全局“^||a”及其后代。它不查找全“^a”。

## 示例

以下示例显示如何将^\$GLOBAL用作\$DATA、\$ORDER和\$QUERY函数的参数。

### 作为\$DATA的参数

^\$GLOBAL作为\$DATA的参数返回一个整数值，表示指定的全局名称是否作为^\$GLOBAL节点存在。下表显示了\$DATA可以返回的整数值。

Value	Meaning
0	全局名称不存在
1	全局名称是包含数据但没有子代的现有节点。
10	全局名称是没有数据但具有子代的现有节点。
11	全局名称是包含数据的现有节点，并且具有子代。

下面的示例测试当前命名空间中是否存在指定的全局变量：

```
/// d ##class(PHA.TEST.SpecialVariables).GLOBAL()
ClassMethod GLOBAL()
{
    KILL ^GBL
    WRITE $DATA(^$GLOBAL(“^GBL”)),!
    SET ^GBL=“test”
    WRITE $DATA(^$GLOBAL(“^GBL”)),!
    SET ^GBL(1,1,1)=“subscripts test”
    WRITE $DATA(^$GLOBAL(“^GBL”))
}
```

```
DHC-APP>d ##class(PHA.TEST.SpecialVariables).GLOBAL()
0
1
11
```

下面的示例测试user命名空间中是否存在指定的全局变量：

```
/// d ##class(PHA.TEST.SpecialVariables).GLOBAL1()  
ClassMethod GLOBAL1()  
{  
    SET $NAMESPACE="USER"  
    SET ^GBL(1)="test"  
    SET $NAMESPACE="%SYS"  
    WRITE $DATA(^$|"USER"|GLOBAL(^GBL))  
}
```

```
DHC-APP>d ##class(PHA.TEST.SpecialVariables).GLOBAL1()  
10
```

下面的示例测试任何命名空间中是否存在指定的进程私有全局变量：

```
/// d ##class(PHA.TEST.SpecialVariables).GLOBAL2()  
ClassMethod GLOBAL2()  
{  
    SET $NAMESPACE="USER"  
    SET ^|PPG(1)="test"  
    SET $NAMESPACE="%SYS"  
    WRITE $DATA(^$||GLOBAL(^PPG))  
}
```

```
DHC-APP>d ##class(PHA.TEST.SpecialVariables).GLOBAL2()  
10
```

## 作为\$ORDER的参数

`$ORDER(^$|nspc|GLOBAL( globalname),direction)`

`^$GLOBAL`作为`$ORDER`的参数，将排序序列中的下一个或上一个全局名称返回到指定的全局名称。如果`^$GLOBAL`中不存在这样的全局名称节点，`$ORDER`将返回空字符串。

注意：`$ORDER(^$GLOBAL(NAME))`不会从IRISSYS数据库返回%global names。

Direction参数指定是返回下一个全局名称还是返回上一个全局名称。如果不提供方向参数，InterSystems IRIS会将排序顺序中的下一个全局名称返回给您指定的全局名称。

以下子例程搜索当前名称空间，并将全局名称存储在名为global的本地数组中。

```
/// d ##class(PHA.TEST.SpecialVariables).GLOBAL3()  
ClassMethod GLOBAL3()  
{  
    GLOB  
        SET NAME=""  
        WRITE !,"???????", $NAMESPACE  
        FOR I=1:1 {  
            SET NAME=$ORDER(^$GLOBAL(NAME))  
            WRITE !,NAME  
            QUIT:NAME=""  
        }
```

```

        SET GLOBAL(I)=NAME
    }
    WRITE !,"?????"
    QUIT
}

```

```
DHC-APP>d ##class(PHA.TEST.SpecialVariables).GLOBAL3()
```

```

??????? DHC-APP
^%ISCWorkQueue
^%cspSession
^%qCacheMsg
^%qCacheMsgNames
^%qCacheObjectErrors
^%qCacheObjectKey
^%qCacheObjectQualifier
^%qCacheSQL
^%qHTMLElementD
^%qJavaMetaDictionary
^%qMgtPortal.Index
^%qPublicSuffix
^%qStream
^%qcspRule
^A
^AA
<INTERRUPT>Visible+4^%SYS.GD
DHC-APP>

```

### 作为\$QUERY的参数

^\$GLOBAL作为\$QUERY的参数，按排序顺序将下一个全局名称返回到指定的全局名称。如果^\$GLOBAL中不存在这样的全局名称作为节点，则\$QUERY将返回空字符串。

注意：\$QUERY(^\$GLOBAL(NAME))不会从IRISSYS数据库返回%GLOBAL NAMES。

在以下示例中，用user命名空间中存在三个全局变量(^GBL1、^GBL2和^GBL3)。

```

/// d ##class(PHA.TEST.SpecialVariables).GLOBAL4()
ClassMethod GLOBAL4()
{
    NEW $NAMESPACE
    SET $NAMESPACE="USER"
    SET (^GBL1,^GBL2,^GBL3)="TEST"
    NEW $NAMESPACE
    SET $NAMESPACE="%SYS"
    WRITE $QUERY(^$|"USER"|GLOBAL(^GBL1)),!
    WRITE $QUERY(^$|"USER"|GLOBAL(^GBL2))
    NEW $NAMESPACE
    SET $NAMESPACE="USER"
    KILL ^GBL1,^GBL2,^GBL3
}

```

```

DHC-APP>d ##class(PHA.TEST.SpecialVariables).GLOBAL4()
^$|"USER"|GLOBAL(^GBL2)
^$|"USER"|GLOBAL(^GBL3)

```

## 作为MERGE的参数

^\$GLOBAL作为MERGE命令的源参数，将全局目录复制到目标变量。Merge将每个全局名称添加为具有空值的目标下标。下面的示例显示了这一点：

```
MERGE gbls=^$GLOBAL( " " )
ZWRITE gbls
```

```
...
gbls( "^zlgsql" )=" "
gbls( "^zlgtem" )=" "
gbls( "^zlgtem1" )=" "
gbls( "^zlgtem4" )=" "
gbls( "^zlgtemp" )=" "
gbls( "^zlgtemp1" )=" "
gbls( "^zlgtemp3" )=" "
gbls( "^zlgtemp5" )=" "
gbls( "^zlgtmp" )=" "
gbls( "^zlj" )=" "
gbls( "^zll" )=" "
gbls( "^zltmp" )=" "
gbls( "^zmc" )=" "
gbls( "^znum" )=" "
gbls( "^zpeterc" )=" "
gbls( "^zsb" )=" "
gbls( "^zseq" )=" "
gbls( "^zstock" )=" "
gbls( "^ztTmp" )=" "
gbls( "^ztrap1" )=" "
gbls( "^zwb1" )=" "
gbls( "^zwhtmp" )=" "
gbls( "^zx" )=" "
gbls( "^zx1" )=" "
gbls( "^zx2" )=" "
gbls( "^zxdd" )=" "
gbls( "^zyb" )=" "
gbls( "^zyb1" )=" "
gbls( "^zyb2" )=" "
gbls( "^zy1" )=" "
gbls( "^zzTT" )=" "
gbls( "^zzdt" )=" "
gbls( "^zzp" )=" "
gbls( "^zzy" )=" "
gbls( "^zzz" )=" "
```

[#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%85%AD%E7%AB%A0-cach%C3%A9-%E5%8F%98%E9%87%8F%E5%A4%A7%E5%85%A8-global-%E5%8F%98%E9%87%8F>