

Consultar la global ^ERRORS utilizando SQL

Artículo

[Alberto Fuentes](#) · Abr 7, 2021



Lectura de 6 min

[Open Exchange](#)

Consultar la global ^ERRORS utilizando SQL

Hola a todos! Os traigo hoy un ejemplo de código que compartía [Robert Cemper](#) para mostrar por SQL los registros de error almacenados en ^ERRORS.

Este es un ejemplo de código que funciona en Caché 2018.1.3 e IRIS 2020.2
No se mantendrá sincronizado con las nuevas versiones.
Es un ejemplo de código y como tal **no** está soportado por el Soporte de InterSystems

Los errores en IRIS/Caché/Ensemble se registran entre otros en la global ^ERRORS. Como este mecanismo se remonta a muchas versiones atrás (décadas del milenio anterior!) su estructura está lejos de las estructuras de almacenamiento de SQL típicas.

El global se escribe mediante la rutina ^%ETN.int y el contenido se hace visible desde la línea de comandos del terminal mediante la rutina ^%ERN o en el Portal de Administración como *Log de Errores de la Aplicación*.

Por defecto no está disponible a través de SQL ya que no hay ninguna clase que lo presente como tabla. Por varias razones:

- Cuando se diseñó, era una buena práctica tener estructuras similares a índices en los mismos globals que los datos. Si digo "similares", significa que no sirve para SQL.
- El contenido de los objetos va a niveles más profundos que el resto. En consecuencia, la profundidad de los subíndices (normalmente IdKey) varía de 3 a 11.

^ERRORS es independiente en cada namespace.

Está estructurado por Day, SequenceByDay, Type, ItemName (Variable, OREF), Value.

zrcc.ERRORStack muestra esta información como tabla SQL.

El contenido más profundo de los objetos se hace visible por la consulta personalizada incluida.

El procedimiento SQL **zrcc.ERRORStack_Dump(Day, Sequence)** devuelve todo el contenido disponible y presenta subíndices y valores como se ve en la lista de globals.

A continuación veremos cómo sacar partido de estas utilidades:

Primero: localiza el día que te interese y el número de secuencia con la ayuda de SQL

Por ejemplo: `SELECT * FROM zrcc.ERRORStack where item='$ZE'`

| Day | Seq | Stk | Type | Item | Value |
|------------|-----|-----|------|------|--|
| 2020-07-02 | 1 | 0 | V | \$ZE | <WRITE>zSend+204^%Net.HttpRequest.1 |
| 2020-07-07 | 1 | 0 | V | \$ZE | <WRITE>zSend+204^%Net.HttpRequest.1 |
| 2020-07-15 | 1 | 0 | V | \$ZE | <WRITE>zSend+204^%Net.HttpRequest.1 |
| 2020-07-20 | 1 | 0 | V | \$ZE | <LOG ENTRY> |
| 2020-07-26 | 1 | 0 | V | \$ZE | <WRITE>zSend+204^%Net.HttpRequest.1 |

A continuación, llama al procedimiento en SQL: CALL zrc.ERRORStack_Dump('2020-07-26',1)

```

Row count: 541 Performance: 0.026 seconds 6557 global references
Ref          Value
2020-07-26,1,"*STACK",0,"V","Routine") zSend+204^%Net.HttpRequest.1
2020-07-26,1,"*STACK",1,"I") 1^S^^^0^
2020-07-26,1,"*STACK",1,"L") 1 SIGN ON
2020-07-26,1,"*STACK",1,"S")
2020-07-26,1,"*STACK",1,"T") SIGN ON
2020-07-26,1,"*STACK",1,"V","%dsTrackingKeys","N","""Analyzer""") 6
2020-07-26,1,"*STACK",1,"V","%dsTrackingKeys","N","""Architect""") 7
2020-07-26,1,"*STACK",1,"V","%dsTrackingKeys","N","""DashboardViewer""") 8
2020-07-26,1,"*STACK",1,"V","%dsTrackingKeys","N","""ResultSet""") 9
2020-07-26,1,"*STACK",1,"V","%objcn") 2
2020-07-26,1,"*STACK",3,"V","Task") &lt;OBJECT REFERENCE>[1@%SYS.Task]
2020-07-26,1,"*STACK",3,"V","Task","OREF",1) 142
2020-07-26,1,"*STACK",3,"V","Task","OREF",1,0) 3671
2020-07-26,1,"*STACK",3,"V","Task","OREF",1,1) +----- general informatio
n -----
2020-07-26,1,"*STACK",3,"V","Task","OREF",1,2) | oref value: 1
2020-07-26,1,"*STACK",3,"V","Task","OREF",1,3) | class name: %SYS.Task
2020-07-26,1,"*STACK",3,"V","Task","OREF",1,4) | %%OID: $lb("13","%SYS.Task")
2020-07-26,1,"*STACK",3,"V","Task","OREF",1,5) | reference count: 5
2020-07-26,1,"*STACK",3,"V","Task","OREF",1,6) +----- attribute values -
-----
2020-07-26,1,"*STACK",3,"V","Task","OREF",1,7) | %Concurrency =
2020-07-26,1,"*STACK",3,"V","Task","OREF",1,8) 4 &lt;Set>
- - -
2020-07-26,1,"*STACK",3,"V","Task","OREF",1,53) | EmailOutput =
2020-07-26,1,"*STACK",3,"V","Task","OREF",1,54) 0
2020-07-26,1,"*STACK",3,"V","Task","OREF",1,55) | EndDate =
2020-07-26,1,"*STACK",3,"V","Task","OREF",1,56) ""
2020-07-26,1,"*STACK",3,"V","Task","OREF",1,57) | Error =
2020-07-26,1,"*STACK",3,"V","Task","OREF",1,58) "&lt;WRITE>zSend+204^%Net.HttpRequest
.1"
2020-07-26,1,"*STACK",3,"V","Task","OREF",1,59) | Expires =
- - -
2020-07-26,1,"*STACK",3,"V","Task","OREF",1,111) | Status =
2020-07-26,1,"*STACK",3,"V","Task","OREF",1,112) "0 "$_lb($lb(5002,"POST to Server
Failed",,,,,,$lb(",%SYS",$lb("$^zSend+204^%Net.HttpRequest.1 +1","$^zPost+1^%Net.H
ttpRequest.1 +1","$^zSendData+20^FT.Collector.1 +1","$^zTransfer+12^FT.Collector.1 +1
","$^zOnTask+3^%SYS.Task.FeatureTracker.1 +1","D^zRunTask+74^%SYS.TaskSuper.1 +1","$^
zRunTask+54^%SYS.TaskSuper.1 +1","D^zRun+26^%SYS.TaskSuper.1 +1"))),$lb(6085,"ISC.Fea
tureTracker.SSL.Config","SSL/TLS error in SSL_connect(), SSL_ERROR_SSL: protocol erro
r, error:14090086:SSL routines:ssl3_get_server_certificate:certificate verify failed"
,,,,,$lb(",%SYS",$lb("e^zSend+303^%Net.HttpRequest.1^1","e^zPost+1^%Net.HttpRequest
.1^1","e^zSendData+20^FT.Collector.1^1","e^zTransfer+12^FT.Collector.1^1","e^zOnTask+
3^%SYS.Task.FeatureTracker.1^1","e^zRunTask+74^%SYS.TaskSuper.1^1","d^zRunTask+54^%SY
S.TaskSuper.1^1","e^zRun+26^%SYS.TaskSuper.1^1","d^^^0")))/ * ERROR #5002: Cache erro
r: POST to Server Failed- ERROR #6085: Unable to write to socket with SSL/TLS configu
ration 'ISC.FeatureTracker.SSL.Config', error reported 'SSL/TLS error in SSL_connect(
), SSL_ERROR_SSL: protocol error, error:14090086:SSL routines:ssl3_get_server_certifi
cate:certificate verify failed' */
2020-07-26,1,"*STACK",3,"V","Task","OREF",1,113) | SuspendOnError =
2020-07-26,1,"*STACK",3,"V","Task","OREF",1,114) 0
2020-07-26,1,"*STACK",3,"V","Task","OREF",1,115) | Suspended =

```

Consultar la global ^ERRORS utilizando SQL

Published on InterSystems Developer Community (<https://community.intersystems.com>)

```
-- --
2020-07-26,1,"*STACK",6,"V","Status1") 1
2020-07-26,1,"*STACK",6,"V","Task") &lt;OBJECT REFERENCE>[1@%SYS.Task]
2020-07-26,1,"*STACK",6,"V","Task","OREF",1) 142
2020-07-26,1,"*STACK",6,"V","Task","OREF",1,0) 3671
2020-07-26,1,"*STACK",6,"V","Task","OREF",1,1) +----- general informatio
n -----
2020-07-26,1,"*STACK",6,"V","Task","OREF",1,2) | oref value: 1
2020-07-26,1,"*STACK",6,"V","Task","OREF",1,3) | class name: %SYS.Task
2020-07-26,1,"*STACK",6,"V","Task","OREF",1,4) | %%OID: $lb("13","%SYS.Task")
2020-07-26,1,"*STACK",6,"V","Task","OREF",1,5) | reference count: 5
2020-07-26,1,"*STACK",6,"V","Task","OREF",1,6) +----- attribute values -
-----
2020-07-26,1,"*STACK",6,"V","Task","OREF",1,7) | %Concurrency =
2020-07-26,1,"*STACK",6,"V","Task","OREF",1,8) 4 &lt;Set>
2020-07-26,1,"*STACK",6,"V","Task","OREF",1,9) | DailyEndTime =
2020-07-26,1,"*STACK",6,"V","Task","OREF",1,10) 0
2020-07-26,1,"*STACK",6,"V","Task","OREF",1,11) | DailyFrequency =
2020-07-26,1,"*STACK",6,"V","Task","OREF",1,12) 0
2020-07-26,1,"*STACK",6,"V","Task","OREF",1,13) | DailyFrequencyTime =
2020-07-26,1,"*STACK",6,"V","Task","OREF",1,14) ""
2020-07-26,1,"*STACK",6,"V","Task","OREF",1,15) | DailyIncrement =
2020-07-26,1,"*STACK",6,"V","Task","OREF",1,16) ""
2020-07-26,1,"*STACK",6,"V","Task","OREF",1,17) | DailyStartTime =
-- --
2020-07-26,1,"*STACK",12,"V","%00000","N","""JournalState""") 12
2020-07-26,1,"*STACK",13,"I") 13^Z^ETNERRB^%ETN^0
2020-07-26,1,"*STACK",13,"L") 13 ERROR TRAP S $ZTRAP="ETNERRB^%ETN"
2020-07-26,1,"*STACK",13,"S") S $ZTRAP="ETNERRB^%ETN"
2020-07-26,1,"*STACK",13,"T") ERROR TRAP
```

541 row(s) affected

[#Consejos y trucos](#) [#Error Handling](#) [#Globals](#) [#Mapeo](#) [#ObjectScript](#) [#SQL](#) [#Caché](#) [#Ensemble](#) [#InterSystems IRIS](#)
[Compruebe la aplicación relacionada en InterSystems Open Exchange](#)

10 2 0 0 26

Log in or sign up to continue
Añade la respuesta

URL de fuente: <https://es.community.intersystems.com/post/consultar-la-global-errors-utilizando-sql>