Article
Robert Cemper · Nov 21, 2020 3m read

ObjectScript over ODBC

Every now and then you may encounter a situation where for various reasons
ODBC is the only option to access a remote system. Which is sufficient as long as you need to examine or change tables.
But you can't directly execute some commands or change some Global.
Special thanks @Anna Golitsyna for inspiring me to publish this.

This examples provides 3 Methods projected as SQLprocedure that enable this if other ways of access are blocked.
Typically by some firewall.

- SQLprocedure **Ping()** returns Server::Namespace::$ZV and allows to check the connection
- SQLprocedure **Xcmd(<commandline>,<resultvar>)** executes the command line you submit and returns the result that you deposit in a variable that you named.
- SQLprocedure **Gset(<global>,<subscript>,<value>,<$data>)** allows you to set or delete a global node
  <global> is a GlobalName in the remote namespace including leading carret; e.g. '^MyGlobal' (sql quoted!)
  <subscript> stands for the complete subscript including parenthesis ;e.g. '(1,3,"something",3)' (sql quoted!)
  <$data> controls if you set the Global Node or execute a ZKILL on it; e.g. 1, 11 to set, 0,10 to ZKILL As you may guess by the name this is especially useful during a Global copy.

The procedure Gset is designed to make use of Global Scanning described earlier.
Combined, they allow a Global copy across any ODBC connection.

Installation:

- On the remote system you need the class provided with this article in OpenExchange
- On the local (source) system you need to define the procedures as Linked SQL Procedures
  SMP>System>SQL> Wizards>Link Procedure
  at that time you local package name is defined (in the examples I used **zrccEX**)
- If you want to run the Global copy you also need to install the Global Scanning class from OEX
  (It is just for comfort)

Examples:

```
USER>do $system.SQL.Shell()
SQL Command Line Shell
[SQL]USER>>select rccEX.Ping()
Expression_1
cemper9::CACHE::IRIS for Windows (x86-64) 2020.1 (Build 215U) Mon Mar 30 2020 20:14:3
3 EDT
```

Check existence of Global ^rcc

```
[SQL]USER>>select rccEX.Xcmd('set %y=$d(^rcc)','%y')
ok: 10
```

Set some value to ^rcc4(1,"demo",3,4)

```
[SQL]USER>>select rccEX.Gset('^rcc4','(1,"demo",3,4)','this is a demo',1)
Expression_1
ok: ^rcc4(1,"demo",3,4)
```

Do a global copy from ^rcc2 to ^rcc4.
First show ^rcc2

```
USER>>select reference,value,"$DATA" from rcc_G.Scan where rcc_G.scan('^rcc2',4)=1
Reference       Value   $Data
 ^rcc2                  10
(1)             1       1
(2)             2       11
(2,"xx")                10
(2,"xx",1)      "XX1"   1
(2,"xx",10)     "XX10"  1
(2,"xx",4)      "XX4"   1
(2,"xx",7)      "XX7"   1
(3)             3       1
(4)             4       11
(4,"xx")                10
(4,"xx",1)      "XX1"   1
(4,"xx",10)     "XX10"  1
(4,"xx",4)      "XX4"   1
(4,"xx",7)      "XX7"   1
(5)             5       1
16 Rows(s) Affected
```

Now run the copy to remote global

```
[SQL]USER>>select rccEX.Gset('^rcc4',reference,value,"$DATA")  from rcc_G.Scan where
rcc_G.scan('^rcc2',4)=1
Expression_1
ok: ^rcc4
ok: ^rcc4(1)
ok: ^rcc4(2)
ok: ^rcc4(2,"xx")
ok: ^rcc4(2,"xx",1)
ok: ^rcc4(2,"xx",10)
ok: ^rcc4(2,"xx",4)
ok: ^rcc4(2,"xx",7)
ok: ^rcc4(3)
ok: ^rcc4(4)
ok: ^rcc4(4,"xx")
ok: ^rcc4(4,"xx",1)
ok: ^rcc4(4,"xx",10)
ok: ^rcc4(4,"xx",4)
ok: ^rcc4(4,"xx",7)
ok: ^rcc4(5)
 16 Rows(s) Affected
```

GitHub

#Other

Source URL:https://community.intersystems.com/post/objectscript-over-odbc