

## Tech Note 10

### Check OPC Server For Memory Leaks

Terravic Research, Inc.

#### Introduction

Memory leaks are very hard to track, as they may only be detectable over a long period of system usage. A systematic and repeatable test setup may be used to track even the smallest memory leaks. Creating and destroying OPC resources in a continuous loop will be used to accomplish this task. Memory status will be checked before and after the test is performed to get the overall memory usage signature. Additionally, memory may be checked during the test execution to monitor the intermediate memory resource usage. The server and client must be running on the same computer for this test to be conclusive.

#### Platform Requirements

Visual OPCTest® Client 8.0.0000 or later.

Examples used in this Tech Note are based on the OPC Foundation sample servers. In order to successfully follow all the procedures outlined in this Tech Note, please download the OPC Foundation's Data Access sample server and the latest Visual OPCTest Client application. If the OPC Foundation sample server is not available, please substitute with another Data Access server and its tag definition. The substitution of server and tag definition in the sample code should be self-explanatory.

#### Basic Steps

The first step consists of connecting to an OPC Data Access server, creating one private group, and adding one tag to the group. In the next step, the above OPC resource allocation is encapsulated in a `FOR` loop to control the number of repetitions. Once the OPC resources are allocated, added tag is read from device, and then all the resources are destroyed, before the loop begins again. Finally, the whole process is preceded with a call to get memory status before the loop begins and after it terminates. All the data is placed in the *ScriptOut* view.

#### Step 1 - Connect to the OPC Data Access Server and Allocate Server Resources

Connecting to OPC Data Access server and adding server resources is as easy as generating a connection script. In the following lines of code we connect an OPC Data Access Server, create one private group, and add a single item to the group:

```
REM connect to data access server
ServerID = OPC.ConnectServer("OPCSample.OpcDaServer.1", FALSE,
"ALL", "")
If ServerID=0 then Exit Sub

REM add private group to opc server
GroupID = OPC.AddGroup(ServerID, "Group_", TRUE, 1000, 2)
If GroupID=0 then Exit Sub

REM add item
ItemID1 = OPC.AddItem(GroupID, "Analog Types/Double", "")
```

## Step 2 – Create a For loop

Encapsulate server connection commands from *Step 1* above inside a `For` loop. Once server resources are created, item will be read (from `DEVICE`) and then all the resources will be destroyed before another loop iteration.

```
MaxLoop = 10000

For Index = 1 to MaxLoop

    REM show visual queue as to which loop is executing
    OPC.StatusPaneMsg "Create Server " & CStr(Index) & " / " &
    CStr(MaxLoop)

    CODE FROM STEP 1 HERE...

    REM read item from device
    Value = OPC.ReadItem( ItemID1, TRUE )

    REM 2 sec wait time before releasing servers
    OPC.UpdateMainWindow
    OPC.Wait 2000

    REM release all the servers and their resources
    OPC.DisconnectAllServers
    OPC.UpdateMainWindow

Next
```

## Step 3 – Get Memory Status

Before and after the `For` loop executes, memory status will be obtained. At the end of the test script, we will compute the memory usage difference (both physical and virtual). All the memory status data will be placed in the *ScriptOut* view.

Get memory status before the loop execution. Store the data in local variables.

```
call OPC.GetMemoryStatus( MemoryLoad, TotalPhys, AvailPhys,
TotalPageFile, AvailPageFile, TotalVirtual, AvailVirtual)
AvailPhysBefore = TotalPhys - AvailPhys
AvailVirtualBefore = TotalVirtual - AvailVirtual
```

Get memory status after the loop execution. Store the data and compute the difference.

```
call OPC.GetMemoryStatus( MemoryLoad, TotalPhys, AvailPhys,
TotalPageFile, AvailPageFile, TotalVirtual, AvailVirtual)
AvailPhysAfter = TotalPhys - AvailPhys
AvailVirtualAfter = TotalVirtual - AvailVirtual
```

```
REM compute the difference
```

```
AvailPhysDiff = AvailPhysAfter-AvailPhysBefore
AvailVirtualDiff = AvailVirtualAfter-AvailVirtualBefore

OPC.ScriptOutputEntry "Memory Status - Difference"
OPC.ScriptOutputEntry " Available Physical Diff: " &
FormatNumber(AvailPhysDiff, 0 , -2, -2, -1)
OPC.ScriptOutputEntry " Available Virtual Diff: " &
FormatNumber(AvailVirtualDiff, 0, -2, -2, -1)
```

## Conclusion

This simple script shows the required setup to collect memory usage of an OPC Data Access server resource allocation and de-allocation. Additional OPC commands may be performed inside the `FOR` loop, such as, activate/deactivate groups/tags, read tag from device/cache, or writing tag data, and other commands.

## Sample Code File

Complete sample code file: `MyServerMemoryTest.otx`  
Sample scripts may be found upon installation of the Visual OPCTest Client software.

## Contact

Contact Terravic with any questions regarding Visual OPCTest Client. Please let us know your concerns, desired improvements, or current issues with our software. We pride ourselves in making changes quickly based on your input. Please contact us at `support @ terravic.com`.

[www.opctest.com](http://www.opctest.com)  
[www.terravic.com](http://www.terravic.com)

