



Cognitier SimpleIPC

Client Code Sample - VBScript

Version 1.0.0.1

April 21, 2009

Copyright © 2009 Cognitier, Inc. All rights reserved.

Cognitier and the Cognitier logo are trademarks of Cognitier, Inc. All other company and product names referred to may be trademarks of their respective owners. This documentation is copyrighted material and is intended exclusively for use by licensed users of Cognitier software. The information in this document is subject to change without notice.

Client Code Sample - VBScript

The following sample client code is provided to demonstrate the use of the most common SimpleIPC client API calls in a VBScript programming environment. The code would be similar for other languages capable of using a COM API. In this example, your application is the client, and it is written in VBScript. You want to invoke a custom server program you recently wrote to be hosted by SimpleIPC.

Notable example situations where you might use the COM API are VBA programming for Microsoft Office applications and classic ASP programming. In the following sample code, a reference to the SimpleIPC COM object is obtained. Thereafter, arguments are passed via the COM object to a remote object running in an IPC server, and the remote object is instructed to run a custom piece of code that you, the software developer, would have previously written. You may have any of several reasons for not wanting the code in question to run in your client's process space. When the call returns, there is an array of output values that was prepared by the remote procedure call. The sample code invokes the remote routine several times in a loop for demonstration purposes. It also illustrates checking the return value from each call to see if there was a failure in the remote process. If so, the code obtains a new session (presumably with a new remote process) and resumes processing.

```
'Declare variables
dim oComp
dim sSessionID
dim iRejectCode
dim sAppID
dim sRoutine
dim oInputArray
dim oOutputArray
dim oErrorArray
dim oWarningArray
dim iElemCount
dim bRet
dim iNumCallsToMake
dim iCallCount
dim bIncrementCallCount
dim SESSIONINVALIDORTIMEOUT
dim SERVERCOMMUNICATIONERR
dim ERRRESUMINGSESSION

'Initialize variables
```

```

sAppID = "App1"
sRoutine = "HelloWorld"
sSessionID = ""
iRejectCode = 0
iNumCallsToMake = 3
iCallCount = 0
bIncrementCallCount = true
SESSIONINVALIDORTIMEOUT = "12001"
SERVERCOMMUNICATIONERR = "12004"
ERRRESUMINGSESSION = "12005"

'Initialize input arguments array
redim oInputArray(2)
oInputArray(0) = "Sample Input 1"
oInputArray(1) = "Sample Input 2"
oInputArray(2) = "Sample Input 3"

'Get reference to COM object
Set oComp = CreateObject("CTOutProcCOMInst1.CTSessionComp1")
WScript.Echo("Obtained reference to CTSessionComp1")

'Make call to obtain a session
bRet = oComp.ObtainSession(sSessionID, sAppID, iRejectCode)
If bRet = false Then
    WScript.Echo("ObtainSession return = " & bRet & "; rej code = " &
    CLng(iRejectCode))
Else
    WScript.Echo("Session obtained; session id = " & sSessionID)
    'For demonstration purposes, make several routine calls in a loop
    Do while iCallCount < iNumCallsToMake
        bIncrementCallCount = true
        'Make call to BasicRoutineCall1
        bRet = oComp.BasicRoutineCall1(sSessionID, sAppID, sRoutine,
oInputArray, oOutputArray, oErrorArray, oWarningArray)
        If bRet = false Then
            If UBound(oErrorArray) >= 0 Then
                If oErrorArray(0) = SESSIONINVALIDORTIMEOUT Or
oErrorArray(0) = SERVERCOMMUNICATIONERR Or oErrorArray(0) = ERRRESUMINGSESSION
Then
                    'The call to BasicRoutineCall1 failed, but the error
code indicates there might be a problem with the session, so try to get a new
session

```

```

        sSessionID = ""
        iRejectCode = 0
        bRet = oComp.ObtainSession(sSessionID, sAppID,
iRejectCode)

        If bRet = false Then
            Exit Do
        Else
            bIncrementCallCount = false
        End If
    Else
        Exit Do
    End If
Else
    Exit Do
End If
Else
    iElemCount = 0
    Do while iElemCount <= UBound(oOutputArray)
        WScript.Echo("Call " & iCallCount & "; Output Array Elem
" & iElemCount & " = " & oOutputArray(iElemCount))
        iElemCount = iElemCount + 1
    Loop

    iElemCount = 0
    Do while iElemCount <= UBound(oErrorArray)
        WScript.Echo("Call " & iCallCount & "; Error Array Elem "
& iElemCount & " = " & oErrorArray(iElemCount))
        iElemCount = iElemCount + 1
    Loop

    iElemCount = 0
    Do while iElemCount <= UBound(oWarningArray)
        WScript.Echo("Call " & iCallCount & "; Warning Array Elem
" & iElemCount & " = " & oWarningArray(iElemCount))
        iElemCount = iElemCount + 1
    Loop
End If

    'Increment the call count unless the previous call failed and we had
to get a new session
    If bIncrementCallCount = true Then

```

```
        iCallCount = iCallCount + 1
    End If
Loop

If iCallCount < (iNumCallsToMake - 1) Then
    WScript.Echo("Unable to complete all routine calls")
End If

'Release the session
oComp.ReleaseSession(sSessionID)
End If

'Dispose of objects
Set oInputArray = Nothing
Set oOutArray = Nothing
Set oComp = Nothing

WScript.Echo("Execution complete")
```