BRS
BIO RECOGNITION SYSTEMS

TECHNICAL TIPS

PRODUCT: BioLock

Subject: Script outputs from BioLocks

Date: 13th December 2007
Revision: 2

BioLock Windows software from version 56 onwards has the ability to perform a logical output on successful verification of a user's finger by means of a script output.

This is perhaps most useful for integrating with home automation systems or for integration with other software (using a BioLock for logical access rather than physical access).

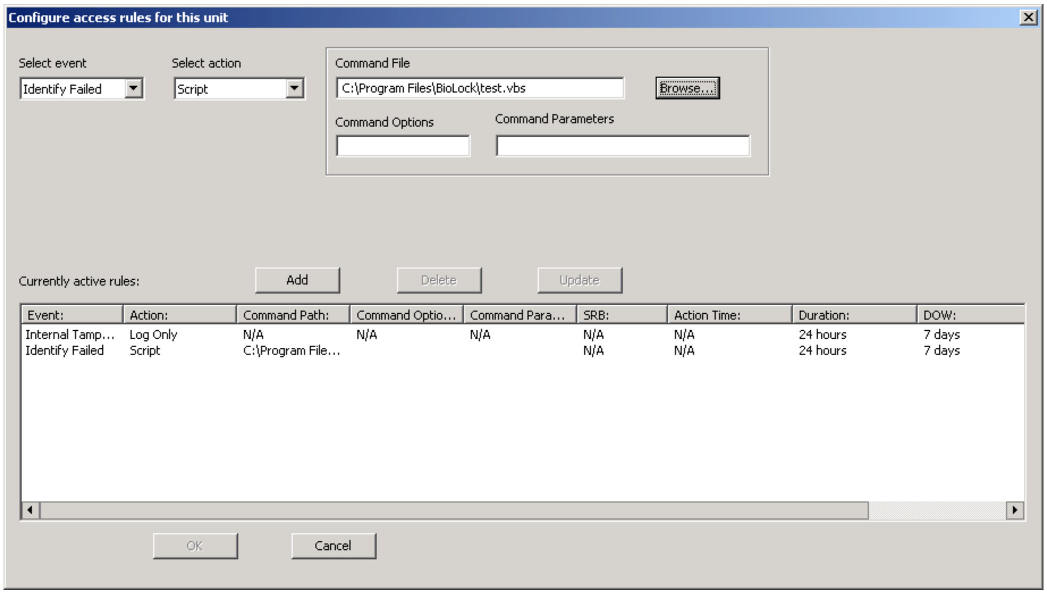
There are two places in the application to use scripting.

Firstly, for anonymous actions or those related to the unit hardware, scripts can be run on events such as "Identify failed", a tamper switch transition, a timezone transition (either for a door hold-open or a REX function timezone), a REX event, a unit connection/ disconnection or the SIOB relay board going on/offline.

To enable this scripting, in the Units screen right-click on a unit on which you wish to run a script output and "configure access rules". Add a rule that has "Script" as an action corresponding to the event you wish. You can add multiple script rules, and you may also want to delete any existing rules that provide a physical action.

The script (including path) you wish to run should then be entered in the Command File field, and any options and parameters you wish to pass to the script entered in the appropriate fields. See later for an explanation of the parameters that are passed to the script.

Author	Andy Saunders
Record ID	TT012071213002.as



Configure access rules for this unit

Select event: Identify Failed

Select action: Script

Command File: C:\Program Files\BioLock\test.vbs

Command Options:

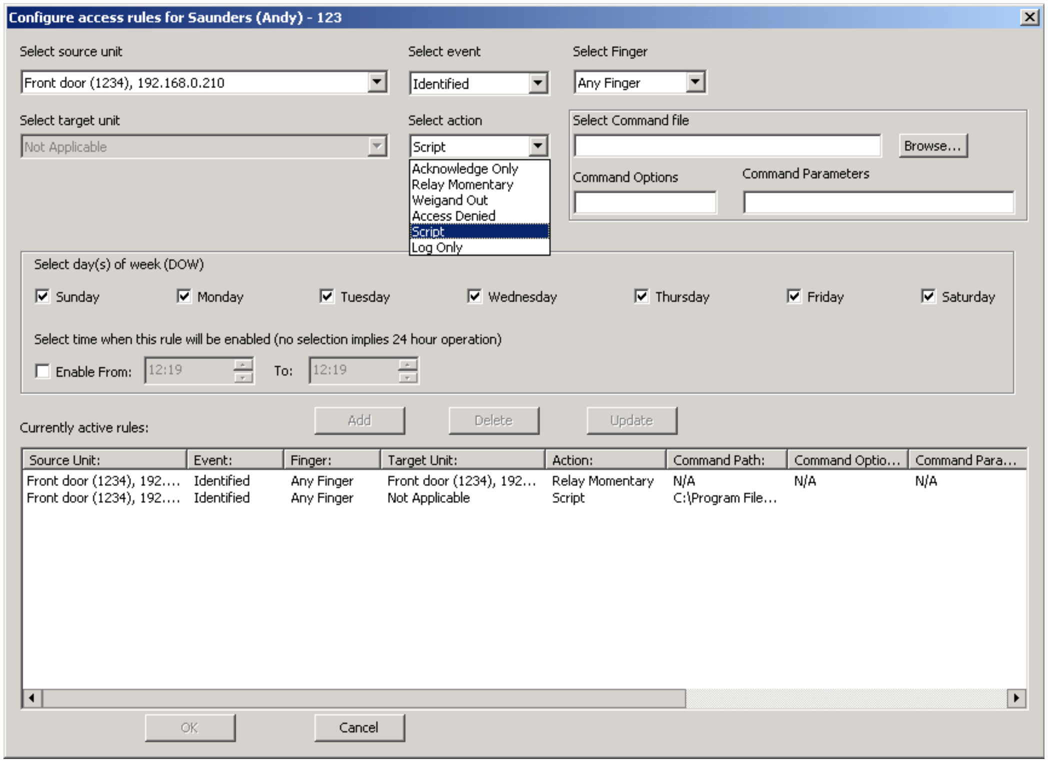
Command Parameters:

Currently active rules:

Event:	Action:	Command Path:	Command Optio...	Command Para...	SRB:	Action Time:	Duration:	DOW:
Internal Tamp...	Log Only	N/A	N/A	N/A	N/A	N/A	24 hours	7 days
Identify Failed	Script	C:\Program File...			N/A	N/A	24 hours	7 days

If setting up multiple units for which you wish to run the same script(s), right-click instead on “Configure Units” and select “Configure Default Access Rules for All Units”. Adding scripting rules here means that any new units added would inherit these default rules (to save having to set up each unit individually).

Secondly, for identification events (successfully recognising a user’s finger), in the users sceen right-click on a user and select “Configure Access Rules”.



Configure access rules for Saunders (Andy) - 123

Select source unit: Front door (1234), 192.168.0.210

Select event: Identified

Select Finger: Any Finger

Select target unit: Not Applicable

Select action: Script

Select Command file:

Command Options:

Command Parameters:

Select day(s) of week (DOW)

☑ Sunday ☑ Monday ☑ Tuesday ☑ Wednesday ☑ Thursday ☑ Friday ☑ Saturday

Select time when this rule will be enabled (no selection implies 24 hour operation)

☐ Enable From: 12:19 To: 12:19

Currently active rules:

Source Unit:	Event:	Finger:	Target Unit:	Action:	Command Path:	Command Optio...	Command Para...
Front door (1234), 192....	Identified	Any Finger	Front door (1234), 192....	Relay Momentary	N/A	N/A	N/A
Front door (1234), 192....	Identified	Any Finger	Not Applicable	Script	C:\Program File...		

There is a third place that scripting can be used. To enable this, the config.ini file has to be edited (usually found in Program Files\BioLock and can be edited with Wordpad or another text editor), to add the following lines:

```
[Enrollment Scripting]
ScriptFile = P:\MTMatcher\Debug\Test.vbs
ScriptOptions =
ScriptCommands =
```

These ScriptOptions and ScriptCommands are the same values you would use in the other scripting features in the other parts of the application. The event is "Enrollment".

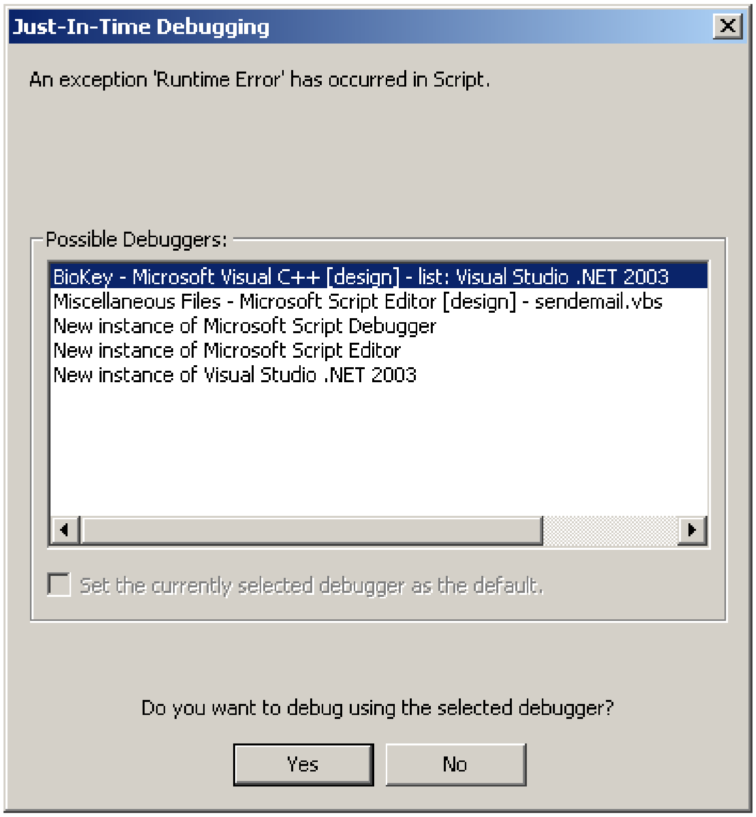
The script gets fired once for each template that has been enrolled when the "Ok" or "Apply" button is pressed on the User screen.

You don’t have to use a script file as the Command file. You can use any Windows executable file – you could choose a JavaScript file or a plain old “*.exe”. Note that the VBScript examples all assume that your PC has Windows Script Host version 5.6 installed (this is the default). You can tell the version installed by typing (in a DOS box): Cscript

All scripts/executables will be passed the event characteristics on the command line. There are at least 9 system level parameters supplied by the BioLock application. Some of these parameters may contain spaces. If so, the relevant parameter will be enclosed in double quotes (""). If the system parameter is not applicable to the event that fired this script then the parameter value will be set to NotApplicable.

The system parameters will be: /YYYY-MM-DD /HH:MM:SS /EventName /UserID /UserName /FingerName /SourceUnitID /SourceUnitDescription /SourceUnitIPAddress

You can also supply your own parameters from the “Command Parameters” edit field in the dialogue box. User-supplied parameters appear after the system parameters described above. If they contain spaces then they must be enclosed in double quotes. Each parameter must be delimited (preceded) by a '/' character.



An exception 'Runtime Error' has occurred in Script.

Possible Debuggers:

BioKey - Microsoft Visual C++ [design] - list: Visual Studio .NET 2003
Miscellaneous Files - Microsoft Script Editor [design] - sendemail.vbs
New instance of Microsoft Script Debugger
New instance of Microsoft Script Editor
New instance of Visual Studio .NET 2003

□ Set the currently selected debugger as the default.

Do you want to debug using the selected debugger?

The “Command Options” field can be used to trigger a script debugger if you have one installed. When an event triggers this script the user will be prompted with a choice of debugger to run the script through. See left – this is very useful for debugging scripts. You can use Visual Studio or you can download the free Windows Script debugger from the internet. To trigger the Windows Script Host debugger the Command Option is: //X

A sample script is included below, which if inspected could be used as a template for development of your scripts, or to understand the command parameters and options (this test script is included on the CD-ROM included with BioLock version 58 and greater):

```
'********************************************************************
'* Function: intParseCmdLine
'*
'* Purpose:  Parses the command line arguments to the variables
'*
'* Input:
'*  [out]    strMachine         machine to query events from
'*  [out]    strUserName        user name to connect to the machine
'*  [out]    strPassword        password for the user
'*  [out]    arrFilters         the array containing the filters
'*  [out]    strFormat          the display format
'*  [out]    strRange           the range of records required
'*  [out]    blnVerboseDisplay  flag to verify if verbose display is needed
'*  [out]    blnNoHeader        flag to verify if noheader display is needed
'*  [out]    objLogs             to store all the given logfles
'* Output:   Returns CONST_PROCEED, CONST_SHOW_USAGE or CONST_ERROR
'*           Displays error message and quits if invalid option is asked
'*
'********************************************************************
Private Function intParseCmdLine( ByRef fp_strDate,                  _
                                  ByRef fp_strTime,                  _
                                  ByRef fp_strEvent,                 _
                                  ByRef fp_strUserID,                _
                                  ByRef fp_strUserName,              _
                                  ByRef fp_strFingerName,            _
                                  ByRef fp_strSourceUnitID,          _
                                  ByRef fp_strSourceUnitDescription, _
                                  ByRef fp_strSourceUnitIP)

    ON ERROR RESUME NEXT
    Err.Clear

    Dim strBLSystemArg  ' to temporarily store the BL system given arguments
to script
    Dim strTemp         ' to store temporary values
    Dim intArgIter      ' to count the number of arguments given by user
    Dim intArgLogType   ' to count number of log files specified - Used in
ReDim
    Dim intFilterCount  ' to count number of filters specified - Used in
ReDim

    strBLSystemArg  = ""
    intArgLogType    = 0
    intFilterCount   = 0
    intArgIter       = 0

    ' Retrieve the command line and set appropriate variables
Do While intArgIter <= Wscript.arguments.Count - 1
  strBLSystemArg = Wscript.arguments.Item(intArgIter)

  IF   Left( strBLSystemArg,1) = "/"  OR    Left( strBLSystemArg,1) = "-"
Then
    strBLSystemArg = Right( strBLSystemArg,Len(strBLSystemArg) - 1  )

    Select Case intArgIter

      Case 0
        fp_strDate = strBLSystemArg
      Case 1
        fp_strTime = strBLSystemArg
      Case 2
        fp_strEvent = strBLSystemArg
      Case 3
        fp_strUserID = strBLSystemArg
      Case 4
        fp_strUserName = strBLSystemArg
      Case 5
        fp_strFingerName = strBLSystemArg
      Case 6
        fp_strSourceUnitID = strBLSystemArg
      Case 7
        fp_strSourceUnitDescription = strBLSystemArg
      Case 8
        fp_strSourceUnitIP = strBLSystemArg
      Case Else
        'User supplied parameters always come after the system parameters
    End Select

  End IF

  intArgIter = intArgIter + 1
Loop '** intArgIter <= Wscript.arguments.Count - 1

End Function

'*********************************************************************
'* Sub: VBMain
'*
'* Purpose: This is main function to start execution
'*
'*
'* Input/ Output: None
'*********************************************************************
```

```
Sub VBMain()

  ON ERROR RESUME NEXT
  Err.clear

  If (WScript.Arguments.Count = 0) Then
    call MsgBox ("Hello World!", 65, "MsgBox Example")
  Else
    Dim strDate
    Dim strTime
    Dim strEvent
    Dim strUserID
    Dim strUserName
    Dim strFingerName
    Dim strSourceUnitID
    Dim strSourceUnitDescription
    Dim strSourceUnitIP
    Dim intOpCode

    intOpCode = intParseCmdLine(strDate, _
                                strTime, _
                                strEvent, _
                                strUserID, _
                                strUserName, _
                                strFingerName, _
                                strSourceUnitID, _
                                strSourceUnitDescription, _
                                strSourceUnitIP)

    call MsgBox (strDate + " " + strTime + ", " + strEvent + ", " +
strUserID + ", " + strUserName + ", " + strFingerName + ", " +
strSourceUnitID + ", " + strSourceUnitDescription + ", " + strSourceUnitIP,
65, "MsgBox Example")

    WScript.Echo (strDate)
    WScript.Echo (strTime)
    WScript.Echo (strEvent)
    WScript.Echo (strUserID)
    WScript.Echo (strUserName)
    WScript.Echo (strFingerName)
    WScript.Echo (strSourceUnitID)
    WScript.Echo (strSourceUnitDescription)
    WScript.Echo (strSourceUnitIP)

  End If

End Sub

' to include the common module
' Dim component                  ' object to store common module

' Set component = CreateObject( "Microsoft.CmdLib" )

' referring the script host to common module
' Set component.ScriptingHost = WScript.Application

' Calling the Main function
Call VBMain()

' end of the Main
Wscript.Quit(EXIT_SUCCESS)
```

FURTHER INFORMATION:

support@brsgrp.com
Bio Recognition Systems
+61 (0)2 9882 8600

Bio Recognition Systems Pty Ltd is a 100% Australian owned and operated hardware and software developer and manufacturer. Located in Lane Cove, Sydney, Bio Recognition Systems Pty Ltd began by offering its customers software and hardware solutions in 1999. Its leading edge BioMetric technology harnesses the power of the newest technology in fingerprint recognition.

Author	Andy Saunders
Record ID	TT012071213002.as