

Report No:

AN116

Title:

Using the WaitForScript method to remote control the ISP-PRO application

Author:	Date:	Version Number:
John Marriott	4th March 2018	1.09

Abstract:

This application note describes how to insert a **'Programming Script'** from a **'Remote Application'** when controlling Equinox ISP programmers using the Equinox ISP-PRO application.

All rights are reserved. Reproduction in whole or in part is prohibited without the prior written consent of the copyright owner. The information presented in this document does not form part of any quotation or contract, is believed to be accurate and reliable and may be changed without prior notice. No liability will be accepted by the publisher for any consequence of its use. Publication thereof does not convey nor imply any license under patent or other industrial or intellectual property rights



Contents

1.0 Introduction	3
1.1 Overview of Sequel / database control	4
1.2 Advantages of database control method	5
1.3 Further reading	6
2.0 Frequently Asked Questions (FAQ)	7
Q1. What is the purpose of the 'WaitForScript.esf' file ?	7
Q2. How does the 'WaitForScript.esf' file work ?	7
Q3. When should I use the 'WaitForScript.esf' file ?	7
Q4. Where should the 'WaitForScript.esf' file be placed ?	7
Q5. What information is required for the 'WaitForScript.esf' file ?	7
Q6. Can I also pass 'Project Names' at the same time as passing the Script Name ?	8
Q7. Can I pass 'Serial numbers' at the same time as passing the Script Name ?	8
3.0 Making a script file to be inserted	. 10
3.1 Overview	. 10
3.2 Script Connection method	. 10
4.0 Setting up ISP-PRO to run WaitForScript	.11
4.1 Overview	. 11
4.2 Files required or production deployment	. 11
4.3 Deploying the required files on the production PC	. 13
4.4 Setting up ISP-PRO to run the WaitForScript script	. 14
4.5 Setting up ISP-PRO for 'Database control'	. 15
4.6 Starting 'production mode' in ISP-PRO	.16
5.0 Testing Script Insertion	.18
5.1 Overview	. 18
5.2 Control definitions - Master / Slave applications	. 19
5.3 Typical control sequence	. 20
5.4 ISP-PRO - how to tell the script is actually running	.21
5.5 Using the 'Database ID' to find the Inserted RECORD	.23
5.6 Polling the status / progress of the Inserted Script	.24
5.7 Determining whether the Script passed or failed	.24
Appendix 1 - Manual record insertion using ADO Explorer	25
1.0 Overview	25
1.1 Starting the ADOExplorer - database utility	25
1.2 Opening the ProgStatus table	26
1.3 Manually inserting a new RECORD into the ProgStatus table	27
1.4 Posting the new RECORD into the database	.21
Appendix 2 - Sequel Query - manual Insertion	29
1.0 Overview	.29
2.0 Using the ADU Explorer utility	.29
3.0 Executing a Sequer query from the Query Window	. JU
4.0 Finding the latest inserted RECORD in the database	. 31 ວ∢
5.0 Poining the Programmer Status via an SQL command	31



1.0 Introduction

This application note describes how to remote control the operation of the Equinox *ISP-PRO* application from a customer's *'Remote Application'* using a technique called *'Script insertion'* which uses *'Sequel command sequences'* via an *'Interface database'* as the control method.



This technique allows the '*Remote Application*' to trigger the execution of a named '*Programming script*' in ISP-PRO by simply inserting a new record into the ISP-PRO '*Interface Database*'. It is also possible to pass other information e.g. a serial number, calibration data etc via fields in the database for ISP-PRO to then program into the target device.



1.1 Overview of Sequel / database control

The ISP-PRO application uses an 'Interface Database' to sequence all programming actions....



To execute a named **'Programming script (sequence)'** in ISP-PRO, the **'Remote Application'** simply has to insert a **'new record'** into the **'ProgStatus'** field of the **'Interface Database'** with the relevant fields filled in e.g. script name, project name(s) etc.

This functionality allows a **Remote Application** to launch any **'Script File'** found in the default ISP-PRO **\scripts** directory from the **Remote Application** without having to manually select the file in ISP-PRO. This is very useful when the **Remote Application** is to be the main control application and allows ISP-PRO to be a slave application which is not visible to the user. A special script called **'WaitForScript.esf'** must be executed in ISP-PRO to allow the Remote Application to insert a script remotely.



1.2 Advantages of database control method

The 'Interface Database' method of control has the following advantages.....

- Completely independent of '*Microsoft Windows'* version works on any version of Windows as long as either the '*ADO*' or '*ODBC*' database layer is installed.
- No dependency on any Equinox-specific 'dll's' as control is handled via 'SQL commands'.
- Does NOT require any pre-installed components which often only work on a specific version of Windows.
- Works with any control software which can communicate to a database via 'SQL queries'.
- Completely independent of target IC / device manufacturer and type the script and projects handle all the 'IC / device specific' configuration.
- Allows complex control sequences to be independently tested without using the customer's control application so problems can be easily found and fixed without having to debug the customer's application.
- Completely independent of Equinox 'programmer type' the same method can be used to control any Equinox programmer.
- Currently supports control of up to 32 x Equinox programmers connected to the same PC.
- Supports passing of other parameters from the control application to ISP-PRO via fields in the database e.g. barcode values, serial numbers, setup parameters, file names etc



1.3 Further reading

This application note focuses on the generic method of the 'Script insertion' control technique and is applicable to all 'programmable device types' and 'programmer types'. Many references are made to our ISP-PRO control software but it is beyond the scope of this application note to explain how to use the ISP-PRO software. Please refer to further sources of documentation listed below for more detailed information.

Application notes:

Please check below for a list of more specific documentation for specific 'programmable device types'.

Application Note	Title
AN109	Remote Application Control of Equinox ISP Programmers using ISP-PRO Utility
AN156	Remote Control of Z-Wave programming scripts when using ISP-PRO control software

User manuals:

This application note also assumes prior knowledge of how to both set up and use the Equinox ISP-PRO software package. Please refer to the following documentation for further information about the ISP-PRO software.

Manual Ref	Title
	ISP-PRO - User Manual

For more detailed information about how ISP-PRO can be controlled via the *'Interface database'* and also examples of Sequel commands to implement this control, please refer to application note AN109.



2.0 Frequently Asked Questions (FAQ)

The simplest way to explain the user of *the 'WaitForScript.esf'* file is by means of a series of questions.

Q1. What is the purpose of the 'WaitForScript.esf' file ?

The 'WaitForScript.esf' file allows a Remote Application (RA) to execute a specified 'Script File' by passing the name of the required 'Script File' via the progstatus.Script_Name field in the ISP-PRO Interface Database.

Q2. How does the 'WaitForScript.esf' file work ?

This script file simply waits for a valid 'Script Name' to be inserted into the progstatus.Script_Name field in the ISP-PRO 'Interface Database' and then executes the specified script. When the inserted script has finished executing, control is returned back to the 'WaitForScript' script.

Q3. When should I use the 'WaitForScript.esf' file ?

If you are planning to control the programmer(s) from a **Remote Application (RA)** and plan to use more than one '**Standalone Programming Project'** for different devices in a Project Collection, then it is a good idea to start using the '**Script Insertion'** methodology. You would also have to use '**Script Insertion'** if you wish to run different scripts for different actions such as reading from EEPROM, processing data and then writing data back to EEPROM.

Q4. Where should the 'WaitForScript.esf' file be placed ?

The 'WaitForScript.esf' file is installed into the \program files\equinox\isppro_v4\scripts directory by default. You should copy this file into your working \scripts directory when you actually want to use it. This will be the same directory where all your other script (*.esf) files are stored.

Q5. What information is required for the 'WaitForScript.esf' file ?

The 'WaitForScript' script is a very simple script file which is waiting for your Remote Application to insert a **NEW RECORD** with the field **progstatus.Script_Name** set to your Script Name (without any file extension).

e.g. If your script file was called *'my_script.esf'*, then you would insert a **NEW RECORD** into the database with *progstatus.Script_Name = 'my_script'*



Q6. Can I also pass 'Project Names' at the same time as passing the Script Name ?

Yes, you can pass 'Standalone project names' needed by the specified script when you pass the 'Script Name'.

Example of inserting a new record:

Insert a new record with the following values for the Status, Script Name and AutoProgram1 project:

- progstatus.Status = REMOTESTART (or whatever status value is required by the inserted script)
- progstatus.ScriptName = testscript (*** Do NOT use a file extension just the name ***)
- progstatus.AutoProgram1 = ProjectName1
- progstatus.AutoProgram3 = ProjectName2

where:

- The 'Script Name' to be inserted is 'testscript'.
- The 'Programmer Status' to start execution of the script is 'REMOTESTART'.
- The 'AutoProgram Project' to be executed is called 'ProjectNamex'.

In this example, the 'Remote application' is passing 'ProjectName1' via the 'AutoProgram1' field and 'ProjectName2' via the 'AutoProgram3' field

Q7. Can I pass 'Serial numbers' at the same time as passing the Script Name ?

Yes, you can pass any device specific data e.g. a 'Serial number' from your 'Remote Application' via a spare field in the database to the script running in ISP-PRO

Example of inserting a new record:

Insert a new record with the following values for the Status, Script Name and AutoProgram1 project:

- **progstatus.Status = REMOTESTART** (or whatever status value is required by the inserted script)
- progstatus.ScriptName = testscript (*** Do NOT use a file extension just the name ***)
- progstatus.AutoProgram1 = ProjectName1
- WriteData.SerialNumber

where:

- The 'Script Name' to be inserted is 'testscript'.
- The 'Programmer Status' to start execution of the script is 'REMOTESTART'.
- The 'AutoProgram Project' to be executed is called 'ProjectNamex'.
- The 'Serial number' value to be passed is written into the WriteData.SerialNumber field in the database.



In this example, the '*Remote application'* is passing '*ProjectName1*' via the '*AutoProgram1*' field and a '*Serial number*' value via the '*WriteData.SerialNumber*' field



3.0 Making a script file to be inserted

3.1 Overview

It is possible to execute any script file you want to using the *WaitForScript 'Script Insertion'* method. However, care must be taken to make sure that the inserted script is set up correctly otherwise the script insertion process may not work correctly.

Please note:

There are also pre-prepared 'Script files' available from Equinox for programming certain specific devices e.g. Z-Wave. Please contact Equinox for further details.

3.2 Script Connection method

The *WaitForScript* is usually set up to wait for a certain status value to be inserted into the *ProgStatus.Status* field. Once a new record has been inserted with the correct value e.g. *"REMOTESTART"*, then ISP-PRO will start to execute the specified script file.

The *WaitForScript* script handles the connection / disconnection.

It is therefore important that the inserted script has the 'Connection method' either set to 'Insert script' or preferably to 'Global - user defined' - see below.....

Check Database V	alues	Chip Era	ise	Final NV	R Read	Final Read From
Script Tasks	Base	ase Project Database Target Cor		Connect/Disconnect		
Schpertable	babe					
Target Connection	and Disco	nnection Met	thod:			

This means that once the new record has been inserted with the 'Script name', the inserted script will execute without waiting for a second connection method. Otherwise the inserted script will sit and wait for the 'Connection method' in the script to be met....which may be different to the WaitForScript.

Important note:

If you set the inserted script 'connection method' to be 'Global - user defined', then this means that the script can either be selected manually in ISP-PRO ('Soft Connect button' connection - useful for manual testing) or used as an 'Inserted script'. This gives you the flexibility of being able to test your scripts manually before trying to automate the process.

Any pre-compiled scripts which you receive from Equinox should already be set to 'Global - user defined' so you do not have to make any changes.



4.0 Setting up ISP-PRO to run WaitForScript

4.1 Overview

This section describes how to setup ISP-PRO for running the '*WaitForScript*' functionality in a production environment.

4.2 Files required or production deployment

The illustration below shows the '*input files*' required to get the following files are required to get the *WaitForScript*' functionality to work....



The following files are required to get the WaitForScript' functionality to work

1. All user Script files (*.esf)

- These are the actual customer/device specific 'Script Files' which ISP-PRO will execute.
- It is possible to have multiple 'Script Files' as there may, for example, be a different script file for every product or perhaps a 'functional test' and 'final production' script file.

2. The default 'WaitForScript.esf' script file



The latest version of the *WaitForScript file* can be found here:

C:\Program Files (x86)\Equinox\ISPPRO_V4\Scripts and is named: WaitForScript_V2-00_101016.ESF

Please note:

If an external developer has made a software release for you, then the '*WaitForScript*' file may also be in the zip file sent to you by your developer.

- 3. The user 'Project Collection' file (*.ppc)
 - The 'Project Collection' file contains all the 'Standalone Programming Projects' which the script may execute.
- 4. For Z-Wave device programming, you may also need the following files:
 - NVR Parameter File (*.NEF)
 - Production Control File (*.csv)

5.ISP-PRO - Interface database:

• This is an Access database which is installed with the Equinox ISP-PRO installer.



4.3 Deploying the required files on the production PC

The **'required files'** as specified in section 4.2 must be copied into the ISP-PRO **'Scripts Folder'** which you plan to use for this project. This can technically be any folder on your production PC or on a server drive.

Instructions:

1. Copy all the 'required files' into the root of your designated 'Scripts Folder' on the production PC.

- 2. Check that you have the following files in your designated 'Scripts Folder' ...
 - Script files (*.esf)
 - default 'WaitForScript_V2-00_100116.esf' script file
 - 'Project Collection' file (*.ppc)

Important note:

As it is possible to call multiple different '*Inserted scripts*' which may refer to different '*Project Collections*', it is very important that all the .*esf* and *.ppc* files are located in the root of the selected '*Scripts Folder*', otherwise ISP-PRO will not be able to locate the required files.



4.4 Setting up ISP-PRO to run the WaitForScript script

The instructions below explain how to set up ISP-PRO to execute the 'WaitForScript' script file.

Instructions:

- Start the ISP-PRO application
- Log in to ISP-PRO (Click the 'Login' icon and then enter password: equinox)
- Click the **<Setup>** icon
- Select the <Programmer Settings> tab

Distance options	morom	or carri opeacory	0101	a option o	alooal o alingo	1.003594
Programmer Settings	Communications	MUX Options	Zip File	Administrator Options	Barcode Scanning	Custom Bitmaps
-Attached programmer details:						
Detect Programmer(s)						
N						
1	Ilse Broadcast Mode					
Assign a programmer nam	ne and script to each avai	lable programming chann	el			
Channel Name:						
Channel 0						
Address:						
Script File Name:						
WaitFor	Script V2-	00 10101	6.ESF			Browse
		_				
This is the	WaitForScripts	script file.				
Script File Directory:	•					1
X:\Sigm	a\Z-Wave	-SPI-Prog	ram-Ca	alibration_S	2_Security	
						J
This is the	selected 'Script	ts Folder'				
						Close

- Make sure that the 'Script File Location' is pointing to the folder where you have placed all your script (*.esf) and project files (*.ppc, *.prj)
- Select the relevant '**PPM channel'** (if there is more than one channel in use) and then click the <**Browse>** button
- Browse to your 'Scripts folder' and select the script called WaitForScript_V2-00_101016.ESF
- Repeat this procedure for each '**PPM channel'** you require to be controlled from the Remote Application
- Click <Close> to save your settings back to the ISP-PRO config file.



4.5 Setting up ISP-PRO for 'Database control'

The instructions below explain how to set up ISP-PRO to use '*database control*' to trigger the execution of a 'programming script'.

Instructions:

- In ISP-PRO, click the **<Setup>** icon
- Select the <Global options> tab
- From the 'Global connection method' drop down menu, select 'Insert Script'

Database Options	Incremental Repository	Global Options
Global Connection Method		
Insert Script		•

This sets up ISP-PRO to wait for the '*Remote Application*' to insert a '*New Record*' into the database with the '*Progstatus.Status*' field set to '*REMOTESTART*'.

• Click <**Close>** to save your settings back to the ISP-PRO config file.



4.6 Starting 'production mode' in ISP-PRO

You are now ready to start the ISP-PRO 'programming network'. This will force the WaitForScript script to execute. This script will then simply sit there waiting for a 'new RECORD' to be inserted into the 'Interface database'.

Instructions:

1. Click the 'Start Auto' icon on the top ISP-PRO icon bar



- You will be asked if you want to start the 'programming network'.
- Click <**OK**>
- ISP-PRO now switches to '*Production mode*' and the '*WaitForScript*' script file will now be executed.

ile <u>C</u> onfigure <u>A</u> ctions <u>S</u> ecu	ity <u>H</u> elp	
Abort 🤗 Clear Errors ⊘ S	irt Auto 💿 Shutdown 🖻 Project Check 🗟 Setup 🙎 Upload Project 😩 Repository 🐠 Programmer Info 🎕	🔋 Logout 🛛 🧰 Overview 💫 Run Database Explorer 😡 Analyse Database
Parameter	Value	
Programmer Name Script Name	Channel 0 WaitForScript_V2-00_101016.ESF	
Selected Projects		WAITING FO
		Next Database ID
		131
		Script Time: 00:00
Channel Status	Messages	ID
1 IN PROGRES	S Waiting for Database (Got Project List From Programmer)	

• The status message for each selected programming channel will display 'Waiting for database'...

Channel	Status	Messages	ID
1	IN PROGRESS	Waiting for Database (Programmer is Waiting)	



• The 'WaitForScript' file will wait indefinitely for the 'Remote Application' to insert a 'new RECORD' into the 'ProgStatus' field of the database with the 'Status' field set to 'REMOTESTART'.



5.0 Testing Script Insertion

5.1 Overview

It is possible to trigger the execution of a specified 'Script File' in ISP-PRO by using any of the methods listed below...

- 1. Manual RECORD insertion using ADO Explorer manual table entry
- See Appendix 1
- 2. Manual RECORD insertion using ADO Explorer Sequel Command entry
- See Appendix 2
- 3. Automated RECORD insertion using *customer's Remote Application*
- See relevant application note

4. Automated RECORD insertion using Equinox *ConsoleEDS console application* (under development)

- See relevant application note

Please refer to the relevant document for further information.



5.2 Control definitions - Master / Slave applications

The remote control of the ISP-PRO application is set up as follows:

Software application	Master / Slave control	Comment
Customer's Remote Application	MASTER	The customer's 'Remote Application' is the 'MASTER' application.
ISP-PRO	SLAVE	ISP-PRO is the 'Slave' application which is controlled by the 'Remote Application'.
Interface Database	Control medium	The 'Interface database' is used as the 'control medium' between the 'Remote Application' and the ISP-PRO software.

The illustration below shows how the customer's '*Remote Application*' controls the ISP-PRO software via the '*Interface database*'...



Please note:

The actual communications link via the database is performed via the 'database layer' on the host PC and will be using either an ODBC or ADO database link / layer.



5.3 Typical control sequence

A typical 'control sequence' between the '*Remote Application*' and ISP-PRO is shown in the table below....

Step#	Remote Application	ISP-PRO application			
1	No action	ISP-PRO is waiting for 'new RECORD' to be inserted into the ProgStatus.Status field			
2	Inserts 'new RECORD' into the interface database.	 Detects that 'new RECORD' has been inserted. Checks that ProgStatus.Status field is set to REMOTESTART Starts execution of script 			
3	No action	 ISP-PRO is now running the selected 'Script file' ISP-PRO will update the <i>ProgStatus.Status</i> field during the script execution 			
4	Polls the ProgStatus.Status field to check the status (progress) of the script running in ISP-PRO	 ISP-PRO will continue running the script until it has finished. 			
5	Polls the ProgStatus.Status field to check the status (progress) of the script running in ISP-PRO	 When the script finishes, ISP-PRO will set <i>ProgStatus.Status</i> field to either 'PASS - Disconnect' or 'FAIL - Disconnect'. 			
6	Detects that script has finished. Displays / logs the PASS / FAIL result.	 ISP-PRO now resets waiting for the next record to be inserted by the 'Remote Application'. 			
	The sequence then repeats from step #2.				



5.4 ISP-PRO - how to tell the script is actually running

Once you have triggered the 'Script Insertion' using one of the methods described in section 5.1, then ISP-PRO should start to run the inserted 'Script File' - see screenshot below...

<u>File</u> <u>Configure</u> <u>Actions</u> <u>Secu</u>	rity <u>H</u> elp	
🛞 Abort 🥔 Clear Errors 🙆 S	art Auto 🕥 Shutdown 🛛 🗟 Project Check 🛛 🗟 Setup 🔮 Upload Project 🎭 Repository 💿 Programmer Info 🏾 🇞 Login 🗖 Overview 🕃 Run Databas	e Explorer 😡 Analyse Database 🛛 🏅
SIGMA SD3502 -	PROGRAMMING / CALIBRATION SEQUENCE	
Now validating N	/R parameters from read back NVR file	
Parameter	Value	
Programmer Name	Channel 0	DROCRAM
Script Name	Z-Wave_S2_SmartStart_ProgCal_SPI_V2-1_db.ESF Id=144)	PROGRAM
Selected Projects	Calibration Firmware-CALIBRATION 1020	
Selected Projects	Run Calibration Firmware=RUNTARGET 1.0.3.0	
	Production firmware=PRODUCTION	Database ID
		Dutubuse ID
		144
		Script Time: 00:02
Channel Status	Messages	ID
1 IN PROGRES	S Operation: Reading from NVR -	144
	File name desunation: %////K% Reading from 0x000000 to 0x0000FF	
]
	Seport Al	port

The 'Script file' name which you inserted into the database should now be displayed....

SIGMA SD3502 - PROGRAMMING / CALIBRATION SEQUENCE			
Now validating NV	Now validating NVR parameters from read back NVR file		
Parameter	Value		
Programmer Name	Channel 0		
Script Name	Z-Wave_S2_SmartStart_ProgCal_SPI_V2-1_db.ESF Id=144)		
Selected Projects	Calibration Firmware=CALIBRATION_1.0.2.0		
	Run Calibration Firmware=RUNTARGET_1.0.3.0		
	Production firmware=PRODUCTION		



The 'channel icon' should display 'AUTOPROGRAM' and the 'Database ID' for this 'Inserted RECORD' is displayed.....

AUTO PROGRAM
Database ID
144
Script Time: 00:02

You should also see the 'Script time' start to increment.

When the inserted 'Script file' has finished execution, ISP-PRO will then return back to the 'Waiting for Database' screen. It will NOT stop at the PASS / FAIL screen. It is then up to the 'Remote Application' to check whether the script passed or failed.



5.5 Using the 'Database ID' to find the Inserted RECORD

When a **'new RECORD'** is inserted into the database (by ADOExplorer or your Remote Application), the database gives this new record a unique **'Database ID'**. ISP-PRO displays this **'Database ID'**. underneath the 'Channel icon' while the script is actually executing.

So for this particular **RECORD** the 'Database ID' is 144....



This 'Database ID' can then be used look up the relevant RECORD in the database.....

🔯 Open Table - ProgStatus		
Id PPM	Status	Script Name
144 Channel 0	PASS - DISCONNECT	Z-Wave_S2_SmartStart_ProgCal_SPI_V2-1_db.ESF
123 Channel 0	PASS - DISCONNECT	Z-Wave_S2_SmartStart_ProgCal_SPI_V2-1_db.ESF
124 Channel 0	PASS - DISCONNECT	Z-Wave_S2_SmartStart_ProgCal_SPI_V2-1_db.ESF
125 Channel 0	PASS - DISCONNECT	Z-Wave_S2_SmartStart_ProgCal_SPI_V2-1_db.ESF
•		
X:\Sigma\Z-Wave-SPI-Program-Calibration_S2	Record 119 of 137	

We can see that the '*ProgStatus.Status*' field for Record ID 144 is '*PASS - DISCONNECT*'...so this particular execution of the script was successful (no errors).



5.6 Polling the status / progress of the Inserted Script

Once you have triggered the 'Script Insertion' using one of the methods described in section 5.1, then ISP-PRO should start to run the inserted 'Script File'. ISP-PRO will then continually update the current status (progress) of the script execution in the 'ProgStatus.Status' field of the current RECORD.

Depending on the nature of the inserted script, the current status (progress) may change a few times a second or for other operations (e.g. FLASH Area programming) it may remain constant for many seconds.

The customer's 'Remote Application' should poll the **'ProgStatus.Status'** field to find out what stage the current script execution is up to.

Please note:

It is important not to poll the ISP-PRO database too often as this will actually cause both the database layer of the PC and ISP-PRO to start running slower. A suitable recommended 'polling interval' is e.g. once a second.

5.7 Determining whether the Script passed or failed

Once the script has completed its execution, then ISP-PRO will automatically update the **'ProgStatus.Status'** field to either...

PASS - DISCONNECT NOW

- This means that the script execution was successful.

FAIL - DISCONNECT NOW

- This means that the script execution has failed for some reason.

The '*Remote Application*' should poll the '*ProgStatus.Status*' field to find out whether the script has finished execution and what the result of the script execution was. i.e. PASS or FAIL.

ISP-PRO will have automatically gone back to the 'Waiting for Database' state where it is again waiting for a 'new RECORD' to be inserted into the database.



Appendix 1 - Manual record insertion using ADO Explorer

1.0 Overview

This section describes how to manually insert a '*new RECORD*' into the *Interface Database* by manually adding the record into the '*ProgStatus*' table using the '*ADO Explorer*' utility.

1.1 Starting the ADOExplorer - database utility

To test the operation of the script, you can manually insert the relevant data into the database using the **'ADO Explorer'** application as follows:

Click the <ADO Explorer> button to launch ADO Explorer

Eile Configure Actions Security Help		•
🔇 Abort 🖉 Clear Errors 🥥 Start Auto 🔾 Shutdown 🗟 Project Check 🕾 Setup 🗟 Upload Project 😂 Repository 💿 Programmer Info 🗞 Logout 🗐 Overview	😥 Run Database Explorer	ò Analyse Database

You will then be asked to log in to the database...

Login	×
Database <u>P</u> rovider	
Microsoft Jet 4.0 OLE DB Provider	<u>B</u> uild
Data Source X:\Sigma\Z-Wave-SPI-Program-Calibration_S2_Security_April_2017\db\a	Load from UDL
	Save to UDL
C Use Windows NT Integrated Security	
 Use a specific user name and password 	
<u>U</u> ser Name	
Password	
<u> </u>	Cancel

- You do NOT need any 'User name' or 'password' for the default Interface Database.
- Click the *<OK>* button to log in.
- Once logged in, you should see a list of tables present in this database.



1.2 Opening the ProgStatus table

The main table which is used to control all programming operations is called the 'ProgStatus' table.

To open the 'ProgStatus' table so it can be viewed / edited.....

- Click the '+' box next to the 'ADO Connection' in the left-hand tab --> a 'Tables' box appears
- Click the '+' box next to the Tables' box to expand the tree --> a list of available 'user database tables' in the Interface Database is displayed see screenshot below.

🔨 ADO Database Explorer
<u> E</u> ile <u>E</u> dit Ex <u>p</u> ort <u>W</u> indow <u>H</u> elp
] 💁 🕺 🐰 🛍 🛍 🗠 🗅 🕑 🗖 Auto F
 C:\Program Files\Equinox\DB\ac_isppro.mdb User Tables BatchInfo EQReadData EQWriteData OscillatorCalibration ProgInfo Projects ReadData Version Version System Tables Views Stored Procedures

- Click on the 'progstatus' table icon so it is highlighted
- Now right-click the 'progstatus' table icon and then select 'Open' from the drop-down menu.
- → A list of all the fields in the 'progstatus' table is displayed in column format across the screen.
- → You may wish to maximise ADOExplorer so you can see more records on screen at the same time.



Elle Edit Export Window Help	
📙 號 🕺 🐰 🗈 💼 🗠 🗅 🕐 🗖 Auto Refresh every 5 seconds	
C:\Program Files\Equinox\DB\ac_isppro.mdb	
BatchInfo Id PPM Status Script_Name	
🕑 🖅 OscillatorCalibration	
The program is a second s	
Image: The second s	
Views	
E Cored Procedures	
C:\Program Files\Equinox\DB\ac_isppro.mdb	
	>

1.3 Manually inserting a new RECORD into the ProgStatus table

To manually insert a NEW RECORD in this table:

- Click the '+' icon --> a blank line is inserted in the database.
- Place the cursor in the 'Status' field by double-clicking in the field.
- Type in the required status to start the script e.g. **'REMOTESTART'**. (This is the status you declared for the 'Target Connection' in 'Script Builder'
- Tab across the columns to the 'Script_Name' column and enter the name of the script you wish to run e.g. 'testscript' (without quotes and without any *.esf extension)
- Tab across the columns to the **'AutoProgam1'** column and enter the name of the AutoProgram project you wish to execute during the script. This project must be in the Project Collection which is currently uploaded to the programmer FLASH Memory Store.

1.4 Posting the new RECORD into the database

To post this record to the database:

- Click the 'Post Edit' (tick) icon.
- → ISP-PRO should now advance to the next stage of the script and then run the specified AutoProgram1 project.
- If you monitor the *progstatus.status* field using **ADO Explorer**, you should see the status of this field updated during the execution of the script.
- The script execution will either PASS or FAIL.
- If it passes, 'PASS DISCONNECT' will be written to the progstatus.status field.
- If it fails, 'FAIL DISCONNECT' will be written to the progstatus.status field.





- The script will then effectively terminate and control will then return back to the 'WaitForScript' script.
- You can repeat the complete sequence by inserting a **NEW RECORD** again.



Appendix 2 - Sequel Query - manual insertion

1.0 Overview

This section describes how to manually insert a 'Sequel query' using the Equinox 'ADO Explorer' utility.

2.0 Using the 'ADO Explorer' utility

A simple but powerful database utility called 'ADO Explorer' is bundled with the ISP-PRO software installation. This utility can be used to view tables in the database. It also features a 'Query window' which can be used to manually execute 'Sequel queries' in the ISP-PRO database. This is an ideal method for manually testing out the 'Script insertion' process before implementing the same functionality in your 'Remote application'.

Instructions

- Start the ISP-PRO application
- Click the 'Login' icon and then enter the password: 'equinox'
- Click the 'Run database Explorer' icon on the top icon bar

The 'ADO Explorer' utility will now start and the following 'Login screen' will be displayed ...

Login	X
Database <u>P</u> rovider	
Microsoft Jet 4.0 OLE DB Provider	<u>B</u> uild
Data Source X:\Sigma\Z-Wave-SPI-Program-Calibration_August_2016\db\ac_isppro.r	Load from UDL
	Save to UDL
O Use Windows NT Integrated Security	
 Use a specific user name and password 	
User Name	
Password	
	Cancel

• Click the 'OK' button to login (leave the 'User name' and 'Password' fields blank.)



• The 'ADO Explorer' utility will now open the ISP-PRO database and show the 'Query Window' in the right-hand pane of the screen.

💐 ADO Database Explorer	
<u>File Edit Export Window</u>	
] 🗽 🕺 % 🖻 🛍 🗠 C	> Auto Refresh every 5 seconds
E…	Query
 ⊕ - ☐ System Tables ⊕ - ☐ Views ⊕ - ☐ Stored Procedures 	Enter your Sequel query here
	Grid . Messages X:\Sigma\Z-Wave-SPI-Program-Calibration_Au
۲	4

3.0 Executing a 'Sequel query' from the 'Query window'

It is possible to execute a 'Sequel query' directly from the 'Query window'.

Instructions:

1. Type or copy and paste your query into the 'Query window'.

Here is a typical 'Sequel query' to try....

INSERT INTO *ProgStatus* (Status, AutoProgram1) VALUES ('**START', 'PRODUCTION'**)

2. To execute this 'Sequel query', click anywhere on the same line as the query and then click the 'Execute' (green PLAY) icon....

aDO Database Explorer		
<u>File Edit Exp</u> ort <u>W</u> indow		
💁 🛛 X 🖻 🛍 🗠 C	Seconds	Click this icon to execute the selected Sequel query
⊡… 🗿 X:\Sigma\Z-Wave-SPI-Prov ⊕… 🧰 User Tables	🖳 Query	
⊕ 💼 System Tables ⊕ 💼 Views	INSERT INTO ProgStatus (Status,	AutoProgram1) VALUES ('START', 'PRODUCTION')
E 🔁 🔁 Stored Procedures		

3. The 'Sequel query' should now execute.



4. The result of the query should now be displayed in the 'Messages' window - see below ...

C Query
INSERT INTO ProgStatus (Status, AutoProgram1) VALUES ('START', 'PRODUCTION')
The result of executing the above query is shown in the window below
\sim
Executed query 'INSERT INTO ProgStatus (Status, AutoProgram1) VALUES (''START'',''PRODUCTION'')' successfully 1 row(s) affected
Grid Image: Contract of the second

This query should have inserted a '*new record*' into the database and entered the following values into the *ProgStatus* table.....

- progstatus.Status = REMOTESTART
- progstatus.ScriptName = testscript
- progstatus.AutoProgram1 = PRODUCTION

5. Checking the new record has been inserted

To check that new record has actually been inserted, you can open and then look in the *ProgStatus* table. You should see a new record has now been inserted.

4.0 Finding the latest inserted RECORD in the database

This 'new RECORD' entry will automatically be given an ID (called the 'database ID' in ISP-PRO) using an 'autonumber'. It is possible to use this 'database ID' to find the 'new RECORD' entry which you have just inserted. Since the 'autonumber' is simply counting up (incremental), then the easiest way to get the entry you just created is to find the highest 'database ID' in the ProgStatus table.

A typical SQL command to display the highest 'database ID' in the ProgStatus table is....

select max(id) from progstatus;

The **'max ID'** found with this command can then be used to identify the correct RECORD in the database. The **'Remote Application'** can then use this **'max ID'** as the **'current ID'** to then poll the correct database 'Status' field to monitor the progress of the script.

5.0 Polling the 'Programmer Status' via an SQL command

ISP-PRO will provide status information of the programming process using the very same entry you created when starting the process. The *Remote Application* must poll the *Progstatus.status* field at regular intervals in order to find out where the programmer is up to in the script. It is also a good idea to display the status to the user.



A typical SQL command to display the current 'programmer status' is.....

select status from progstatus where id=currentid;

Where 'currentid' is the current 'Database ID" of the record which you have inserted.