



EQTools

Version II

*Project Management Utility
for Equinox Production ISP System*

Getting Started Guide



The Embedded Solutions Company

Contents

Copyright Information.....	iii
Technical Support.....	iv
1.0 Overview	1
1.1 Supported Programmers	1
1.2 Features	1
1.3 PC Operating System (O/S) compatibility	1
1.4 Minimum PC Hardware Requirements.....	2
1.5 PC Software Requirements.....	2
1.6 Associated Software.....	2
2.0 Software Installation.....	3
2.1 Overview	3
2.2 Installation of associated Software Applications.....	3
2.3 Windows 2000/NT/ME/XP considerations.....	3
2.3.1 Overview	3
2.3.2 Installing EQTools under Windows 2000, NT or ME.....	3
2.4 Running the EQTools installation	5
2.5 Installation Directory Structure	13
3.0 Configuring EQTools after installation	14
3.1 Overview	14
3.2 COM (Serial) port selection	15
3.2.1 Selecting the COM port.....	15
3.2.2 Overview of Communications settings.....	16
3.3 Programmer Info	17
3.3.1 Overview	17
3.3.2 Reading the Programmer Info from a programmer.....	18
3.3.3 Overview of programmer License Upgrades	19
3.3.4 Entering a License Upgrade	20
3.4 Selecting an Interface Database within EQTools	22
3.4.1 Overview	22
3.4.2 MDAC Database Driver Error	22
3.4.3 Selecting the required database alias.....	22
3.5 Installation Troubleshooting Guide.....	24
4.0 Getting Started Guide	26
4.1 Overview	26
4.2 Launching EQTools.....	27
4.3 Overview of EQTools Components	30
4.3.1 Overview	30
4.3.2 Launching an EQTools utility	31
4.4 EQTools Toolbar Overview	32
4.4.1 Overview	32
4.4.2 Standard Toolbar	32
4.4.3 Project Builder Toolbar	33
4.4.4 Project Manager Toolbar	35
4.4.5 Equinox Development Suite (EDS) Toolbar.....	38
4.5 Re-opening files	39
4.5.1 Overview	39
4.5.2 Re-opening a previously opened file.....	39
4.5.3 Re-opening all files from last session of EQTools	39
4.6 Uploading a Project Collection to a programmer.....	41
4.6.1 Overview	41
4.6.2 Procedure	42

5.0 Equinox Development Suite (EDS).....	44
5.1 Overview	44
5.2 Features.....	44
5.3 Launching Equinox Development Suite (EDS)	45
5.3.1 To make a new Development Project using the EDS Wizard	45
5.3.2 To test a pre-compiled Programming Project in EDS.....	46
5.4 <Base Project> tab	47
5.4.1 Overview.....	47
5.4.2 Editing programming settings in EDS.....	48
5.5 <Flash / EEPROM> tabs.....	50
5.5.1 Overview.....	50
5.5.2 Buffer Window - Control Buttons	51
5.5.3 File loading into the Buffer Window	53
5.5.4 Automatically reloading a file on change	53
5.5.5 Saving data from the Buffer to file	54
5.5.6 Target Control Buttons	57
5.6 <Fuses> tab	58
5.6.1 Overview.....	58
5.6.2 Fuse Types.....	59
5.6.3 Target Fuses - buttons	60
5.6.4 Local Fuses - buttons	61
5.6.5 Exporting / Importing Fuse Settings to / from File	62
5.7 <Security> tab	63
6.0 Project Builder instructions	67
6.1 Overview	67
6.2 Programming Project Source File (*.ppm) overview	68
6.3 Project Builder – Launching the Wizard.....	70
6.4 Project Builder – Project Tasks Tab.....	71
6.5 Project Builder – Header Information Tab.....	72
6.6 Project Builder – Programmer Type tab.....	74
6.7 Project Builder – Target Device tab	76
6.8 Project Builder – Target Oscillator tab	79
6.9 Project Builder – SPI Settings tab	81
6.10 Project Builder – Target Power Supply Settings tab	84
6.11 Project Builder – Pre-programming State Machine tab.....	86
6.12 Project Builder – Fuse Settings tabs.....	88
6.13 Project Builder – FLASH / EEPROM tab.....	93
6.14 Project Builder – Security Settings tab.....	95
6.15 Project Builder – Saving and compilation.....	96
6.16 Using version control	97
7.0 Project / Script File explanations.....	99
7.1 Overview	99
7.2 EQTools – Source files & output files overview	99
7.3 Files required by ISP-PRO to execute a programming script	101
7.4 Distributing scripts and associated files in a ZIP file	102
7.4.1 Creating a ZIP file for distribution to a Production Facility.....	102
7.5 Testing a script file	103

Copyright Information

Information in this document is subject to change without notice and does not represent a commitment on the part of the manufacturer. The software described in this document is furnished under license agreement or nondisclosure agreement and may be used or copied only in accordance with the terms of the agreement.

It is against the law to copy the software on any medium except as specifically allowed in the license or nondisclosure agreement.

The purchaser may make one copy of the software for backup purposes. No part of this manual may be reproduced or transmitted in any form or by any means, electronic, mechanical, including photocopying, recording, or information retrieval systems, for any purpose other than for the purchaser's personal use, without written permission.

© Copyright 1998 to 2007 - Equinox Technologies UK Limited. All rights reserved.

AtmelTM and AVRTM are trademarks of the Atmel Corporation

Microsoft, MS-DOS, WindowsTM and Windows 95TM are registered trademarks of the Microsoft Corporation

IBM, PC and PS/2 are registered trademarks of International Business Machines Corporation

Intel, MCS 51, ASM-51 and PL/M-51 are registered trademarks of the Intel Corporation

Every effort was made to ensure accuracy in this manual and to give appropriate credit to persons, companies and trademarks referenced herein.

Equinox guarantees that its products will be free from defects of material and workmanship under normal use and service, and these products will perform to current specifications in accordance with, and subject to, the Company's standard warranty which is detailed in Equinox's Purchase Order Acknowledgment.

Technical Support

It is often the case that users experience problems when installing or using a product for the first time. Equinox is unable to answer technical support questions about this product or its use by telephone.

If you have a technical support problem, please consult the following list for help:

i. **Manual**

ii. **Internet Web Site**

Please browse to: www.equinox-tech.com

iii. **E-mail**

Please e-mail any technical support questions about this product to:

support@equinox-tech.com

Equinox will try our best to answer your questions about this product as quickly as possible. However, we cannot promise an immediate reply. Please consult our web site for new software updates as the problem that you are enquiring about may have already been fixed in a new version.

iv. **Fax**

Please fax any technical support questions about this product to: +44 (0) 1942 841975

Equinox will try our best to answer your questions about this product as quickly as possible. However, we cannot promise an immediate reply. Please consult our web site for new software updates as the problem that you are enquiring about may have already been fixed in a new version.

1.0 Overview

These instructions detail how to install and use the Equinox '**EQTools Version 2.xx**' Software Suite. EQTools is a dedicated Windows application used to create and upload 'Programming Projects' to Equinox programmers and also to create and debug 'Programming Scripts' for use with the Equinox ISP-PRO software.

The EQTools software is supplied either on a CD-ROM or it can be downloaded from the Equinox Web Site.

1.1 Supported Programmers

The EQTools software is used to configure/control the following Equinox programmers:

Programmer	Programming channels	Comments
MiniProg	1	Low-cost portable field ISP Programmer
Epsilon5	1	Portable ISP Programmer
FS2000A	1	Portable field / production ISP Programmer
FS2003	1	Portable field / production ISP Programmer
PPM3-MK2	1	Production Programming Module – Version 2 (1 channel)

Please note

The Equinox PRO101 / 104 / 108 and PPM3-MK2 (version1) programmers are not supported by **EQTools Version 2**.

Please carry on using **EQTools Version 1** with these programmers.

1.2 Features

EQTools is used for the following:

- To create 'Programming Projects' (*.ppm / *.prj) for Equinox programmers
- To create 'Project Collections' (*.ppc) which contain the Programming Projects (*.prj)
- To create 'Programming Scripts' (*.esw, *.esf, *.fus) to run within ISP-PRO
- To upload 'Programming Projects' (Project Collections) to the target programmer(s)
- To test / debug 'Programming Scripts' on a single programming channel before running them in ISP-PRO.

1.3 PC Operating System (O/S) compatibility

- This software is a true 32-bit application
- Compatible with Windows 95, 98, NT4 Client, Windows 2000, Windows ME, Windows XP
- It has not been fully tested under Windows Vista

- It will not operate under DOS, UNIX or Windows 3.11

1.4 Minimum PC Hardware Requirements

- 100% IBM compatible PC 486+ (Pentium recommended)
- 10 MB free hard disk space
- Spare PC COM (serial) port

1.5 PC Software Requirements

- **Windows 95, 98, NT4 (client), Windows 2000, Windows ME, Windows XP**
This software has not been fully tested on Windows Vista.
- **COMCT132.dll (Toolbar dll)**
EQTools requires the latest 'Toolbar dll' from Microsoft which is a file called '**COMCT132.dll**'. This file is also required and automatically installed by many other applications including 'Microsoft Internet Explorer 5 (IE5)'. Equinox do not include the COMCT132.dll in the EQTools installation because it is a large file and would add considerably to the download time of the application from our Web Site.

1.6 Associated Software

- **ISP-PRO**
This is a software application designed to control Equinox programmers in a 'production' environment. For further information please download the 'ISP-PRO Getting Started Guide' from our web site.
- **ConsoleEDS**
This software utility allows any Equinox ISP programmer to be controlled via simple command line arguments (DOS Control). For further information please download the 'AN111 - ConsoleEDS User Guide' from our web site.
- **Winzip**
The Winzip utility is very useful for transferring project/script files between the developer and the production facility. This utility must be purchased separately from 'Winzip Computing Inc.' at www.winzip.com.

2.0 Software Installation

2.1 Overview

This installation installs the EQTools application plus associated device libraries and help files. If you are not planning to use the ISP-PRO software to control your programmer(s), then no other software besides EQTools needs to be installed.

2.2 Installation of associated Software Applications

EQTools does not require any other software to be installed by default. If you have the necessary license updates to run ISP-PRO with your programmer, then you will need to install ISP-PRO before installing EQTools. Please refer to the '**ISP-PRO – Software Installation and User Guide**' for further information.

You may also find it useful to install the **Winzip** application for zipping and unzipping files. **Winzip** provides an ideal and secure method of transferring programmer related files between **EQTools** and **ISP-PRO** and between a 'developer' and a 'production facility'.

2.3 Windows 2000/NT/ME/XP considerations

2.3.1 Overview

If you are installing this software under Windows NT, Windows 2000, Windows ME or Windows XP, it is necessary to ensure that whichever user profile the software is installed under has the relevant privileges to write into the **\program files** directory and also the PC registry. Please contact your System Administrator for help on this subject.

Please note

The software installation instructions as detailed in the next section must be carried out while logged in as the 'System Administrator' or a domain user who has full 'administrator' privileges. Failure to do this may cause errors during installation and execution of this software.

2.3.2 Installing EQTools under Windows 2000, NT or ME

Please follow the steps listed below:

1. Log in as the 'System Administrator' or a domain user who has full 'administrator' privileges
2. Make sure that all the '**Domain Users**' who will be using the software have the required file privileges to write files into the **\program files** directory.
3. Make sure that all the '**Domain Users**' who will be using the software have read/write access to the EQTools directory structure (**\program files\equinox**), as this is required by the software at run-time.

4. The easiest way to make sure that all '**Domain Users**' can use the software is to make them all so-called '**Local Administrators**' on your PC.

Please note

If you encounter any problems during installation of this software, please consult your 'IT Administrator' to check that your PC is set up correctly with FULL ADMINISTRATOR PRIVILEGES.

2.4 Running the EQTools installation

1. Close down all other applications which are currently running on your PC
2. Log in as the '**System Administrator**' or as a domain user with administrator privileges
3. Use your file browser to find the EQTools installation software called **eqtools.exe**
The install may be named with a version number eg. **eqtools210_403.exe**.
4. Check that your programmer is supported by **EQTools Version 2**
Please note: The PRO101 programmer is NOT supported by this software.
5. If you already have a version of EQTools installed, you will need to remove this version before installing the new version.

To remove a previous version of EQTools:

- Backup any files you wish to keep in the **\program files\equinox** directory.
- If you are using the database functionality and wish to keep your original database, please back up the following file **\program files\db\ac_isppro.mdb**.
- Launch your PC Control Panel (Select <Start> <Settings> <Control Panel>)
- Select the <Add / Remove Programs> utility
- Scroll down to the EQToolsV2 item and click the <Remove> button
- Wait for confirmation that the software has been removed and then quit the <Add / Remove Programs> utility

6. Double-click the **eqtools.exe** installation file
→ The EQTools installation program will launch and display the screen below.



7. Click <Next>
→ This will display the first screen for the InstallShield Wizard

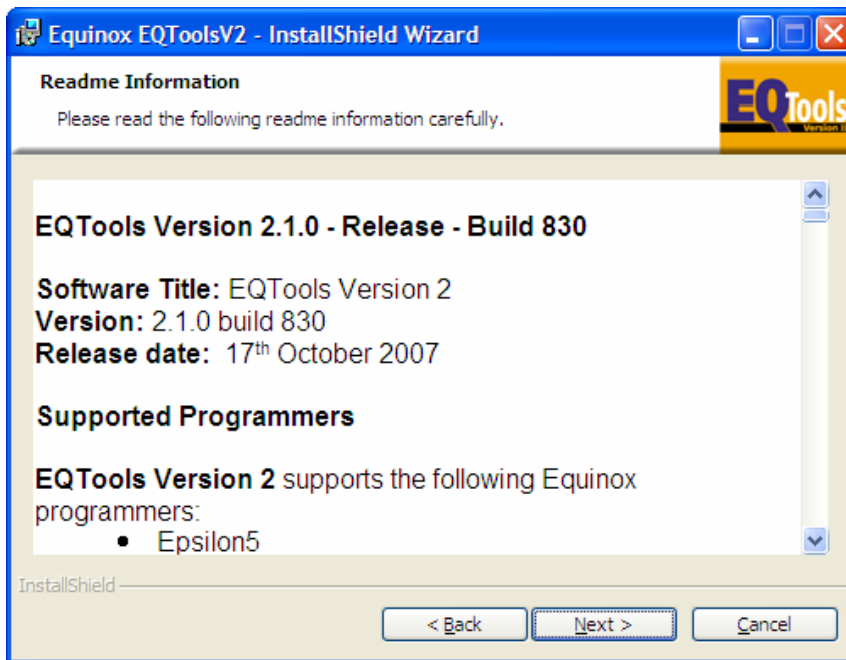
Please note:

If you have a previous version of EQTools installed, the installer will ask you to remove this version before allowing you to proceed. Please follow the instructions detailed in step (5).



8. Click <Next>

→ The InstallShield Wizard now displays the 'Readme Information' screen showing software version number, build, release date and programmers supported by the software etc.



9. Click <Next>

→ The InstallShield Wizard now displays the 'Customer Information' screen.

Please enter your user name, company name in the boxes provided.

The software will be accessible to all users (profiles) on your PC by default. If you wish to limit the use of the software to one specific user, click the '**Only for me (user)**' radio button.

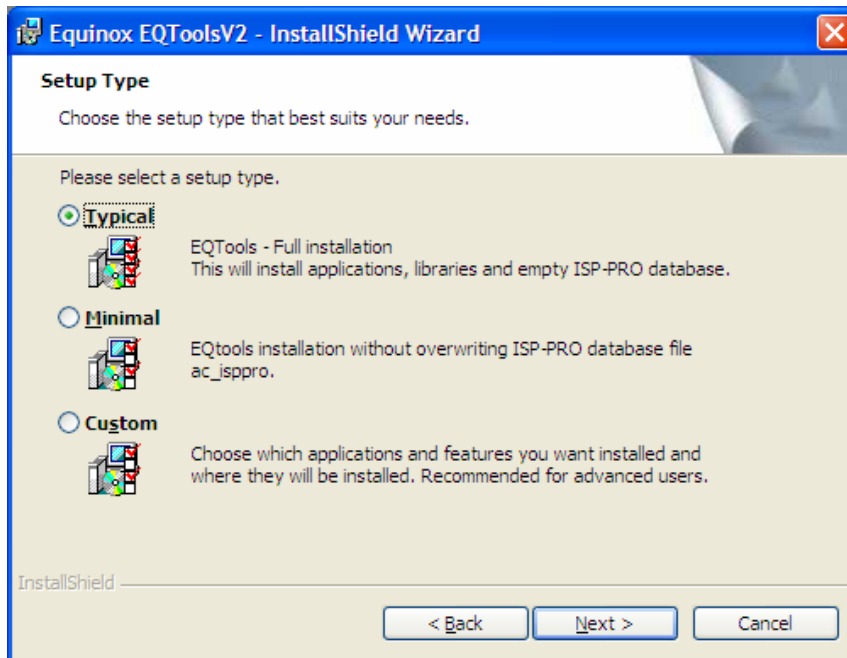
The screenshot shows the 'Customer Information' screen of the 'Equinox EQToolsV2 - InstallShield Wizard'. The window title is 'Equinox EQToolsV2 - InstallShield Wizard'. The main heading is 'Customer Information' with a sub-instruction 'Please enter your information.' Below this, there are two text input fields: 'User Name:' with the text 'User' entered, and 'Organization:' with the text 'Equinox Technologies Ltd' entered. Below the input fields, there is a section titled 'Install this application for:' with two radio button options: 'Anyone who uses this computer (all users)' (which is selected) and 'Only for me (User)'. At the bottom of the window, there are three buttons: '< Back', 'Next >', and 'Cancel'. The 'Next >' button is highlighted with a dashed border.

10. When you have completed the information Click <Next>
→ The InstallShield Wizard now displays 'Destination Folder' screen. To change the folder click <change> and then browse to the required folder for the software to be installed into.

The screenshot shows the 'Destination Folder' screen of the 'Equinox EQToolsV2 - InstallShield Wizard'. The window title is 'Equinox EQToolsV2 - InstallShield Wizard'. The main heading is 'Destination Folder' with a sub-instruction 'Click Next to install to this folder, or click Change to install to a different folder.' Below this, there is a folder icon and the text 'Install Equinox EQToolsV2 to: C:\Program Files\Equinox\'. To the right of this text is a 'Change...' button. At the bottom of the window, there are three buttons: '< Back', 'Next >', and 'Cancel'. The 'Next >' button is highlighted with a dashed border.

11. When you have selected the folder Click <Next>

12. The InstallShield Wizard now displays the 'Setup Type' screen.



Please choose from one of the following installation options:

Typical

- Select this option if you want to install EQTools and all related software.
- This will install EQTools, Device Libraries, ConsoleEDS, Upload Wizard, ADO Explorer, Configit and the Interface Database.
- Any data contained in an existing copy of the Interface Database will be overwritten.
- Click <Next> and proceed to step 14

Minimal

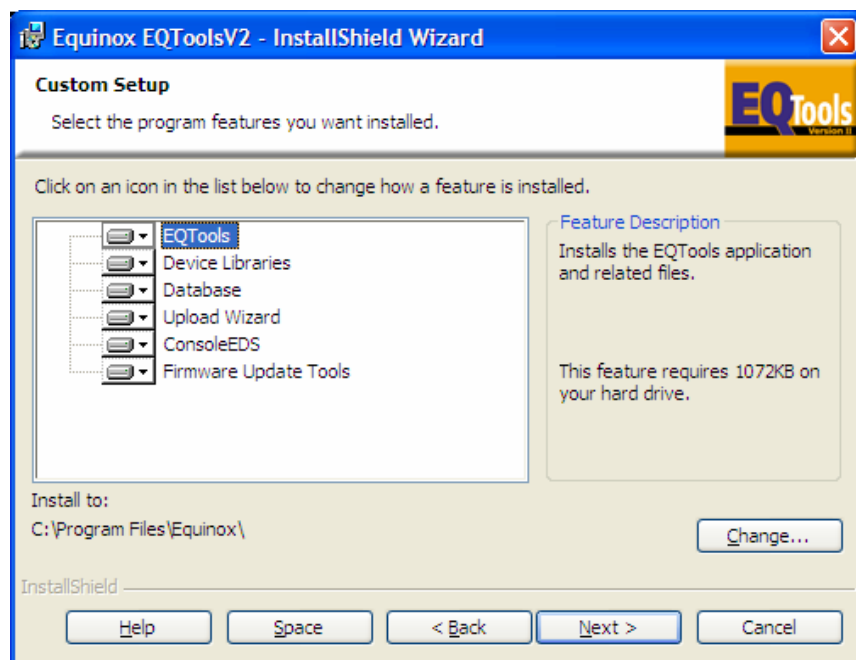
- Select this option if you want to install all software applications but you do not want to overwrite the Interface Database.
- This will install EQTools, Device Libraries, ConsoleEDS, Upload Wizard, ADO Explorer, and Configit.
- The Interface Database will not be over-written so all existing data will be retained.
- Click <Next> and proceed to step 14

Custom

- Select this option if you want to manually select which software utilities are installed.
- Click <Next> and proceed to step 13

13. Custom Installation Setup

The 'Custom Setup' screen allows you to choose which applications and support files are installed onto your PC. This makes it possible to only install eg. Upload Wizard if the software is for a production user who does not need EQTools.

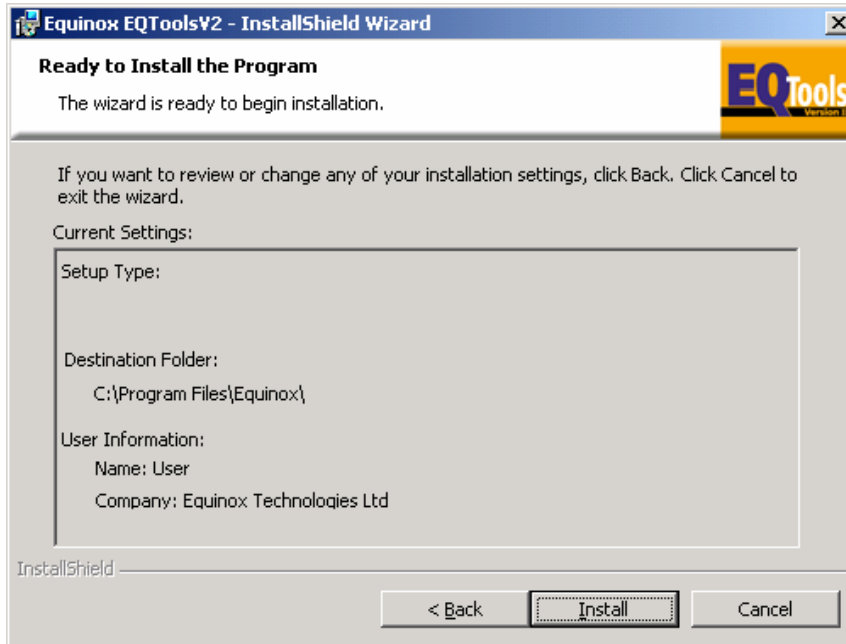


Please note:

- All options are selected to be installed as standard.
- Click on the symbol next to each option in order to select whether it will be installed or not.
- If you are installing EQTools, it is recommended that the EQTools application, Device Libraries and Upload Wizard are all installed at the same time.
- You can install the 'Upload Wizard' utility (used for uploading projects to a programmer) without installing any other software. This is useful for production administrators who want to offer the bare minimum software interface to the production user.
- You can install the 'ConsoleEDS' utility (command line driven control) without installing any other software. However, if the ConsoleEDS script uses the database then it would need to be installed.

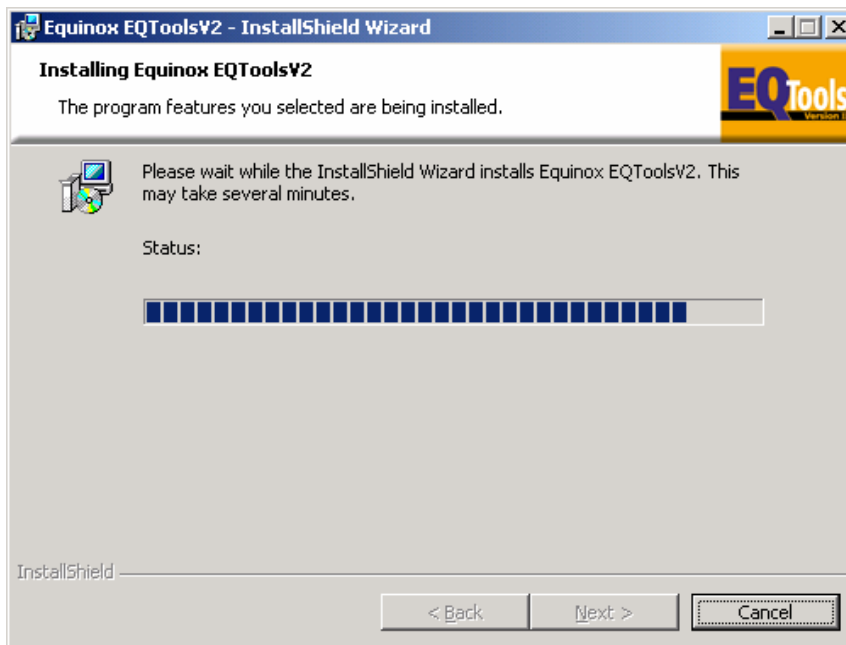
14. Click <Next>

→ The InstallShield Wizard now displays the 'Ready to Install' screen. Please check that all the information displayed is correct before proceeding with the installation.



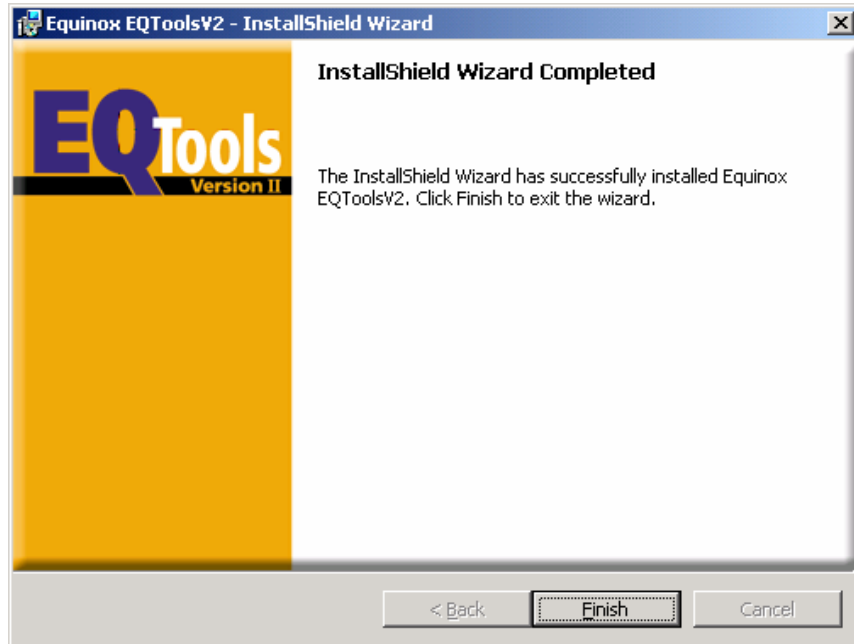
15. Click <Install>

→ The InstallShield Wizard now displays the 'Installing Equinox EQTools' screen. This screen has a status bar to show you what stage the installation process is at.



16. Click <Next>

→ The InstallShield Wizard now displays the 'InstallShield Wizard Completed' screen, showing that the software has been installed.



17. Click <Finish> to exit the Wizard and your software is now ready to use.

2.5 Installation Directory Structure

The default installation directory is: **\\program files\\equinox**

The installation creates the folders listed in the table below:

Main Directory	Sub Directory	Description
\\eqtools		Main applications + associated files <ul style="list-style-type: none"> • EQTools.exe • ConsoleEDS.exe • Default.rst
	\\lib	Device Library Files (*.lib)
\\db		<ul style="list-style-type: none"> • Equinox Interface Database Access 2000 Database – filename: ac_isppro.mdb • ADOExplorer – Database utility
\\UploadWizard		Upload Wizard application files <ul style="list-style-type: none"> • UploadWizard.exe
\\Firmware		<ul style="list-style-type: none"> • Configit – Firmware Update utility • Firmware Update projects for use with Upload Wizard

3.0 Configuring EQTools after installation

3.1 Overview

The EQTools installation establishes the following default settings:

- Expects programmer to be connected to COM1 with a Communications Address of ZERO (0).
- Maintains an ADO Database Connection to the **ac_isppro.mdb** database stored in **Program Files\Equinox\DB\ac_isppro.mdb**. If your PC Windows installation does not have the 'ADO Database Layer (Microsoft MDAC drivers)' installed already then EQTools will give you an error message and it is then necessary to download and install the MDAC Database Drivers from the Microsoft Website.

If you wish to change any of these settings, please refer to the table below which point to the relevant section for further information. To check the configuration of your programmer, please use the 'Programmer Information' utility.

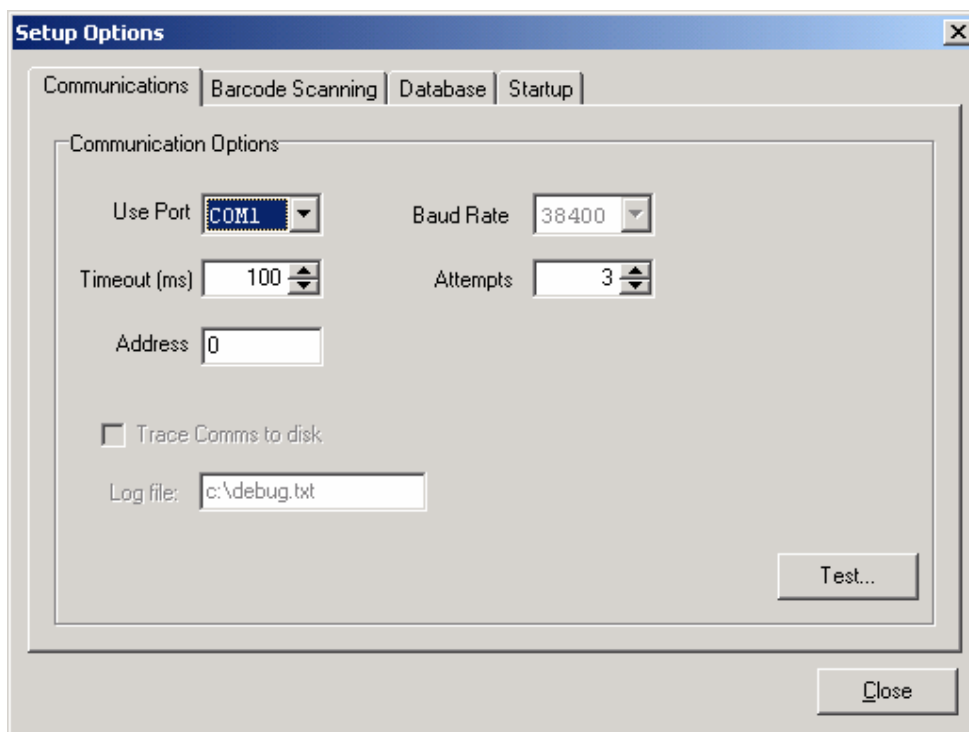
#	Subject	Refer to section
1	COM (Serial) Port Selection	3.2
2	Programmer Information	3.3
3	Selecting an Interface Database within EQTools	3.4
4	Installation Troubleshooting Guide	3.5

3.2 COM (Serial) port selection

The default EQTools installation looks for a programmer to be connected to the PC COM1 serial port. If you are using a different COM port and/or are experiencing a 'Communications Error', it may be necessary to set up the communication parameters.

3.2.1 Selecting the COM port

1. Launch EQTools
2. Select <Programmer><Communications and scripting setup> from the top menu bar
3. Select the <Communications> tab



4. Select the COM port to which the programmer(s) is (are) connected
5. The Baud Rate is fixed at 38,400 and cannot be changed by the user
6. The Communications Address defaults to address '0' which is correct for the Epsilon5 and FS2000A programmers. The PPM3-MK2 programmer can be set to operate from any address between 0 and 15. The corresponding node address must be entered in the 'Address' field.
7. All other settings should not be altered.
8. Click the <Test> button
→ EQTools will attempt to communicate with the attached programmer.

Result	Description
PASS	Message: <i>Communicated with attached programmer successfully.</i>
FAIL	Message: <i>Failed to communicate with attached programmer.</i> Please check the following: <ul style="list-style-type: none">• Programmer is connected to selected COM port• Selected COM port is not in use by another application• Programmer is powered up

9. Click <Close> and your settings will be saved.

3.2.2 Overview of Communications settings

Use Port

Select the PC COM (serial) port to which the programmer is connected.
Please ensure that the chosen COM port is not in use by another application.

Baud Rate

This is the communications speed used by the PC to communicate with the target programmer(s). The baud rate is fixed at 38,400 for the FS2000A and PPM programmers.

Timeout (ms)

This is the maximum time after allowed after a message has been sent from the PC before an acknowledgment from the programmer is received. The default value is 100 ms, which should be fine for most PC's. If you experience unreliable communications between the PC and programmer, try increasing this timeout value in increments of 100 ms.

Attempts

This is the number of times the PC tries to send the same message to a programmer. If the programmer does not respond correctly within the given number of attempts then a 'Communications Error' is reported. The default value is 3 attempts.

Address

This is the physical communications address of the target programmer on the Equinox Production ISP Network. Each programmer (programmer module) must have a unique node address. For an FS2000A or EPSILON5 programmer, this address is fixed at zero. A PPM can have any node address from 0 to 15 (factory default = address 0).

3.3 Programmer Info

3.3.1 Overview

All Equinox programmers feature an electronic 'build tag' which contains unique build information about the programmer. This information is useful for programmer diagnostics and traceability.

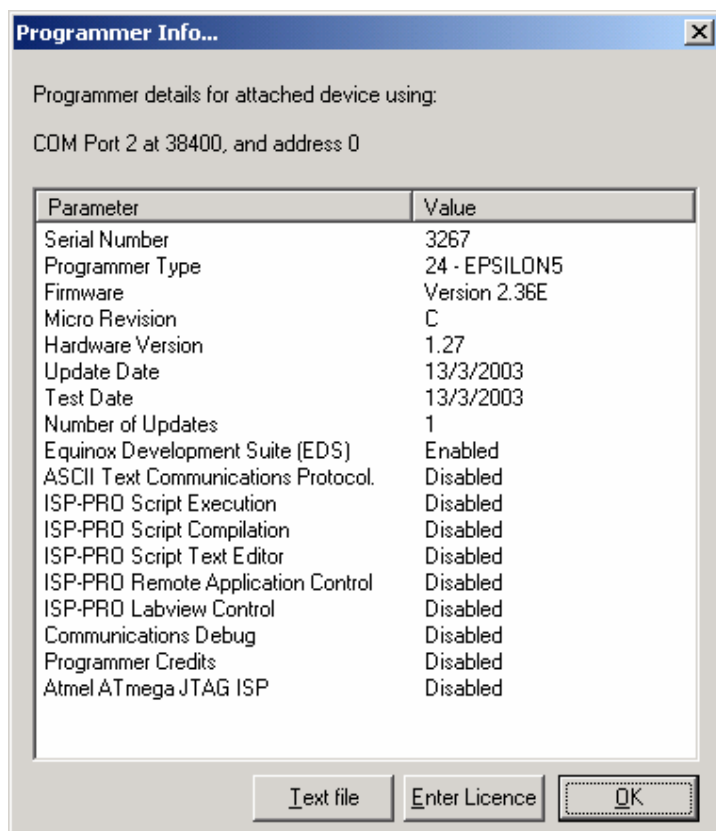
The information stored electronically in the programmer is detailed below:

Parameter	Description
Serial Number	This is a unique serial number for this particular programmer. The serial number should be the same as the one printed on a label on the back of the programmer enclosure and also on the programmer PCB.
Programmer Type	This is the Equinox programmer model. e.g. PPM3-MK2, FS2000A, EPSILON5
Firmware	This is the programmer firmware revision. Equinox update the programmer firmware periodically to add new algorithms. The firmware version will be increased by 0.01 every time the firmware is updated.
Micro Revision	This is the revision of silicon of the programmer 'Supervisor Microcontroller'.
Hardware Version	This is the hardware revision of the programmer. Equinox can trace the original build status of the programmer from this number.
Update date	This is the date on which the programmer firmware was last updated.
Test Date	This is the date on which the programmer was tested.
No of updates	This is the number of times the programmer firmware has been updated since original manufacture.

3.3.2 Reading the Programmer Info from a programmer

To read the 'Programmer Information' from a programmer:

- Connect the programmer to a spare PC COM (serial) port
- Power up the programmer
- Launch EQTools
- Select <Programmer><Programmer Info>
→ The 'Programmer Info' dialog is displayed.



Please note

This screen also displays which options the attached programmer is enabled for. Please see 'License Upgrades' for a full explanation of all these options.

3.3.3 Overview of programmer License Upgrades

It is possible to upgrade your Equinox programmer to support certain features and Device Libraries. These upgrades should be ordered from Equinox. For upgrade prices and availability please refer to the Equinox Price List.

The following table explains the special features of the licence upgrades:

License Upgrade	Description
Equinox Development Suite (EDS)	This is a Development Suite used to control Equinox programmers in a development environment. This feature is available as standard with all programmers.
ASCII Text Communications Protocol	This is a simple ASCII protocol which can be used to control any Equinox programmer from any Remote System which can generate ASCII commands via an RS-232 port.
ISP-PRO Script Execution	This option allows an Equinox programmer to execute 'Programming Scripts' using the ISP-PRO Software Suite.
ISP-PRO Script Compilation	This option allows a Programming Script Wizard file to be compiled within EQTools.
ISP-PRO Script Text Editor	This options allows you to manually edit the textual version of Script File (*.esf). This is only required for experts and so is normally disabled.
ISP-PRO Remote Application Control	This option enables a Remote Application to control the programmer eg. Visual Basic Control Demo or your own control program.
ISP-PRO LabView Control	This option enables the programmer to be controlled via a National Instruments – Labview 5 or 6 control program.
Communications. Debug	This option allows all communications between EQTools and a programmer to be logged to a debug file.
Atmel ATmega JTAG ISP	This option enables a programmer to support JTAG In-System Programming of the Atmel ATmega AVR Microcontroller Family.
ConsoleEDSPRO	Enables the use of ConsoleEDS PRO with this programmer
ConsoleEDS Multi-Port	Enables the use of multiple sessions of ConsoleEDS on the same computer. This allows control of multiple programmers when they are connected to multiple COM ports on the PC.
Standalone Upload Utility	This options enabled the use of the Upload Wizard utility without requiring EQTools to be installed.

3.3.4 Entering a License Upgrade

It is possible to upgrade your programmer to support certain features and device libraries which are not supplied as standard with the programmer.

1. Order the required 'License Upgrade' from Equinox directly or from your local distributor.

The following information is required to process the order

- Order code of the required License Upgrade
- Programmer type eg. FS2000A
- Serial number(s) of the programmer(s) to be upgraded

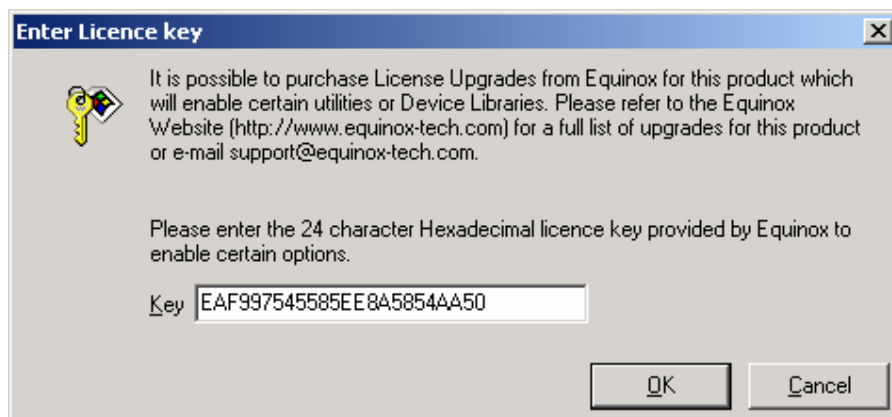
2. Equinox will then send you via e-mail or fax a 'License String' for each feature / library which you have ordered.

This 'License String' is specific to the serial number of the programmer and the feature / upgrade which has been ordered. The string is a 24 character alphanumeric string.

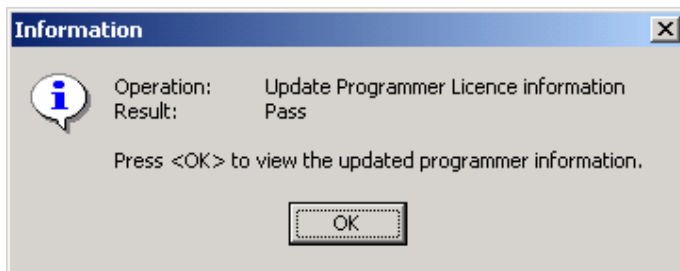
Typical example string: **EAF9A05455A3EE8A5554AA50**

3. To upgrade your programmer using the License String

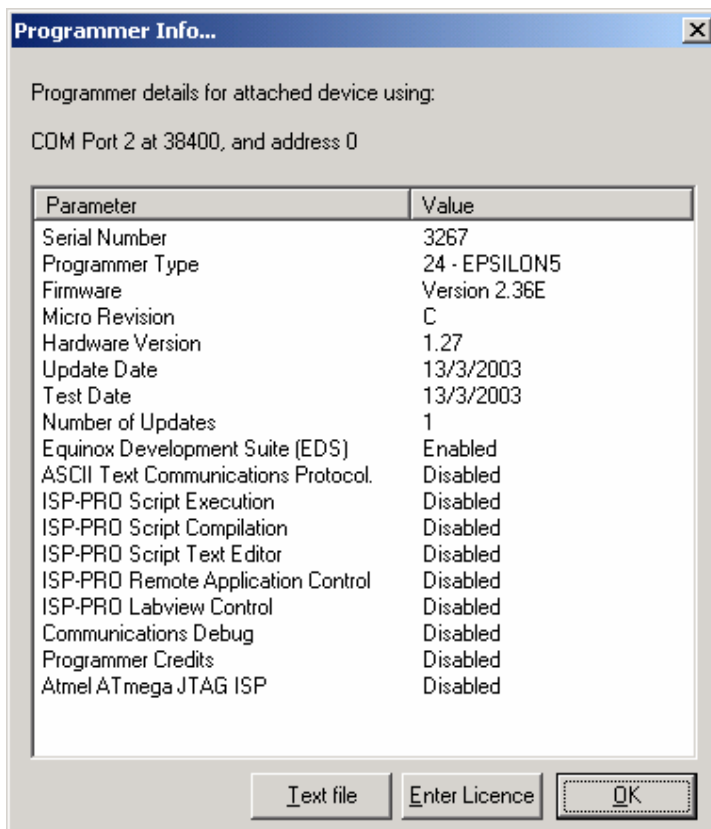
- Connect the programmer to a spare PC COM (serial) port
- Power up the programmer
- Launch EQTools
- Select <Programmer><Programmer Info>
→ The 'Programmer Info' dialog is displayed.
- Press the <Enter License> button
→ The Enter License dialog box is displayed



- Enter the 'License String' in the box provided and then press <OK>.
- EQTools will now try to validate the entered 'License String' with your programmer. If the string is accepted, the following message will be displayed:



- Click <OK> to display the Programmer Information dialog again and then verify that the required feature / option has been enabled.



- Click <OK> to quit out of this screen.
- Your programmer has now been updated.

Please note

If the string is not accepted, please re-check that the 'License String' has been entered correctly and that it is the correct string for the relevant programmer serial number.

3.4 Selecting an Interface Database within EQTools

3.4.1 Overview

The default installation of EQTools installs a so-called 'Interface Database' in the *?;/program files/equinox/eqtools/db directory*.

The default Interface Database is a Microsoft Access 2000 database (**ac_isppro.mdb**). This can be replaced by any other user-defined database which has the required fields and tables. In order for EQTools to connect to this database, a suitable 'ADO Database connection' must be established to the database. This should have been performed by the standard EQTools installation, so you should not have to perform any further setup. If the connection to the database has not been made correctly, you can manually configure it by following the instructions detailed in the next section.

3.4.2 MDAC Database Driver Error

EQTools maintains an ADO Database Connection to the **ac_isppro.mdb** database stored in **Program Files\Equinox\DB\ac_isppro.mdb**. If your PC Windows installation does not have the 'ADO Database Layer (Microsoft MDAC drivers)' installed already then EQTools will give you an error message and it is then necessary to download and install the MDAC Database Drivers from the Microsoft Website. You can find a link to the MDAC page on the Epsilon5 Downloads page on the Equinox Website.

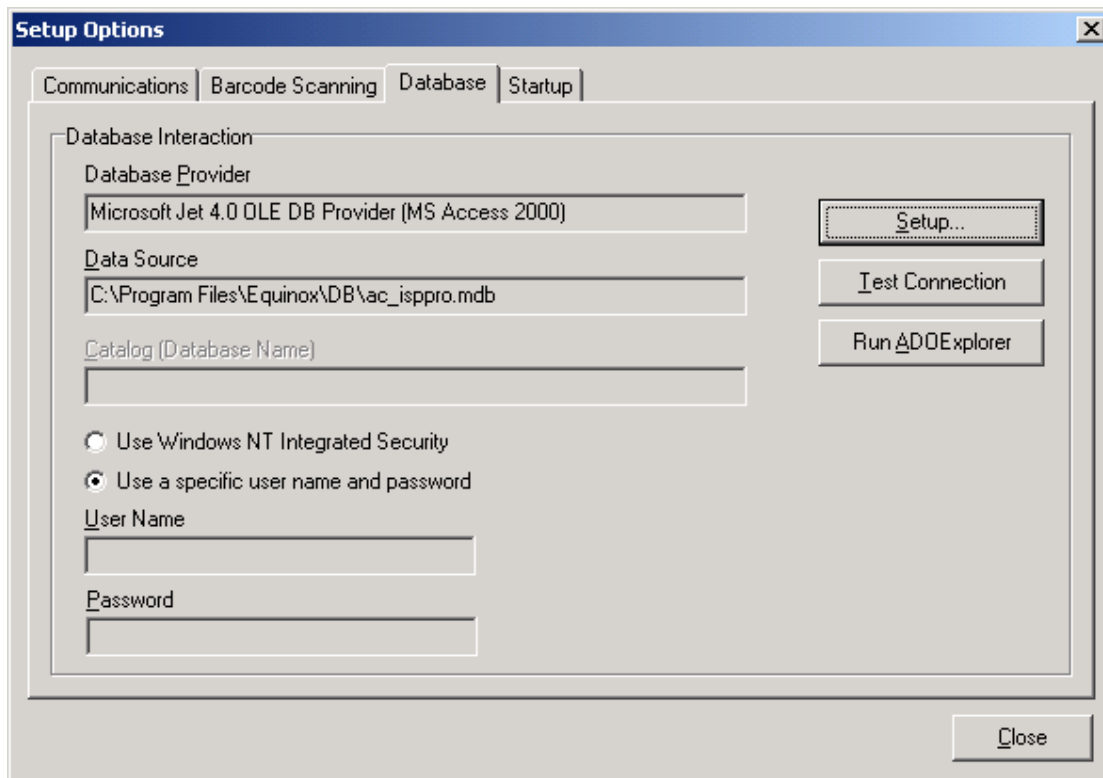
3.4.3 Selecting the required database alias

Once the database alias has been set up, it is then a simple matter of configuring ISP-PRO to use the required alias as follows:

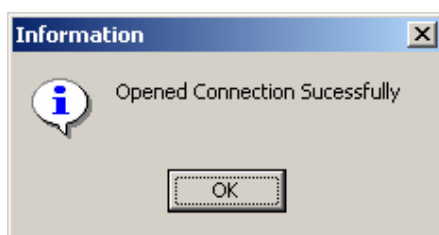
1. Launch the EQTools application
2. Select <Programmer><Communications and Scripting> and then select the <Database> tab

Please note:

- If the <Database> tab is not displayed, then EQTools cannot locate the default Interface Database **ac_isppro**.



3. If you are prompted for a database password for the default databases, simply select <OK> leaving the username and password fields blank.
4. Click <Test Connection> to test whether ISP-PRO is configured so it can connect to the specified Interface Database.
If successful, you should see the following confirmation message:



Click <OK>.

5. Click <Close> to save these database settings
→ EQTools is now configured to use your specified Interface Database
6. If you wish to view the tables within your database, you can use **the 'ADOExplorer'** utility which can be launched by pressing the <ADOExplorer> button in the <Database Options> tab.

3.5 Installation Troubleshooting Guide

The table below details some of the most common errors which may be encountered during the installation of the EQTools software. If you encounter any other errors which are not listed, please e-mail a full error description to support@equinox-tech.com.

Error No	Problem	Overview / Solution
N/A	EQTools installation program fails when initially launched.	<p>Overview</p> <p>This error should only happen when installing under Windows 2000 or Windows NT. It is probable that you do not have the required user privileges to install software into the \program files directory on your PC.</p> <p>Solution</p> <p>Log in to your computer as the 'System Administrator' or as a domain user with administrator privileges. See section 2.3 for further details.</p>
N/A	EQTools Toolbar is not displayed correctly.	<p>Overview</p> <p>EQTools requires the latest 'Toolbar dll' from Microsoft which is a file called 'COMCT132.dll'. This file is also required and automatically installed by many other applications including 'Microsoft Internet Explorer 5 (IE5)'. Equinox do not include the COMCT132.dll in the EQTools installation because it is a large file and would add considerably to the download time of the application from our Web Site.</p> <p>Solution</p> <p>Install the 'COMCT132.dll' using one of the following methods:</p> <p>Run the 'Microsoft Common Controls Update Program' which can be found on the CD-ROM supplied with the programmer or can be downloaded from the Equinox Web Site. The filename of the update program is '40comupd.exe'.</p> <p>OR</p> <p>Install 'Microsoft Internet Explorer 4' or any later version. Installing this application will automatically install the latest version of the COMCT132.dll.</p>

N/A	Some EQTools dialog screens are not properly displayed or are cropped so all the information cannot be read.	<p>Overview</p> <p>Many of the EQTools dialogs display a considerable amount of data on the screen. It is necessary to select 'Small Fonts' mode to properly display these dialogs, otherwise some data may appear off the visible area of the screen.</p> <p>Solution</p> <p>To set your PC to display 'Small Fonts':</p> <ul style="list-style-type: none"> • Select <Start><Settings><Control Panel><Display> • Select the <Settings> tab • Click the <Advanced> button • From the 'Display menu', select the 'Font size' as 'Small fonts'. • Click OK • You will then be asked whether you wish to restart Windows • Click <YES> and wait for Windows to reboot • Your PC is now set up to display in 'Small font' mode.
-----	--	---

N/A = Not applicable (i.e. no error code available)

4.0 Getting Started Guide

4.1 Overview

The EQTools application is an Integrated Development Environment which can be used to:

- Create 'Programming Projects' (*.ppm / *.prj)
- Create 'Project Collections' (*.ppc) which contain the Programming Projects (*.prj)
- Test Programming Projects with the Equinox Development Suite (EDS)
- Create 'Programming Scripts' (*.esw, *.esf, *.fus) to run within ISP-PRO
- Upload 'Programming Projects' (Project Collections) to the programmer(s)
- Test / debug 'Programming Scripts' on a single programming channel before running them in ISP-PRO.

Within EQTools, it is possible to run various utilities including the Project Builder, Project Manager, Project Upload Wizard, Script Builder and Script Debugger. Please see section 4.3 for an overview of these utilities.

This 'Getting Started Guide' is simply a quick-start guide to help get you up-and-running as quickly as possible. For further help on any subject please use the on-line context-sensitive help system by pressing <F1> at any time.

4.2 Launching EQTools

To launch EQTools, select <Start><Program><Equinox><EQTools>

→ The main EQTools window will be displayed as shown below.

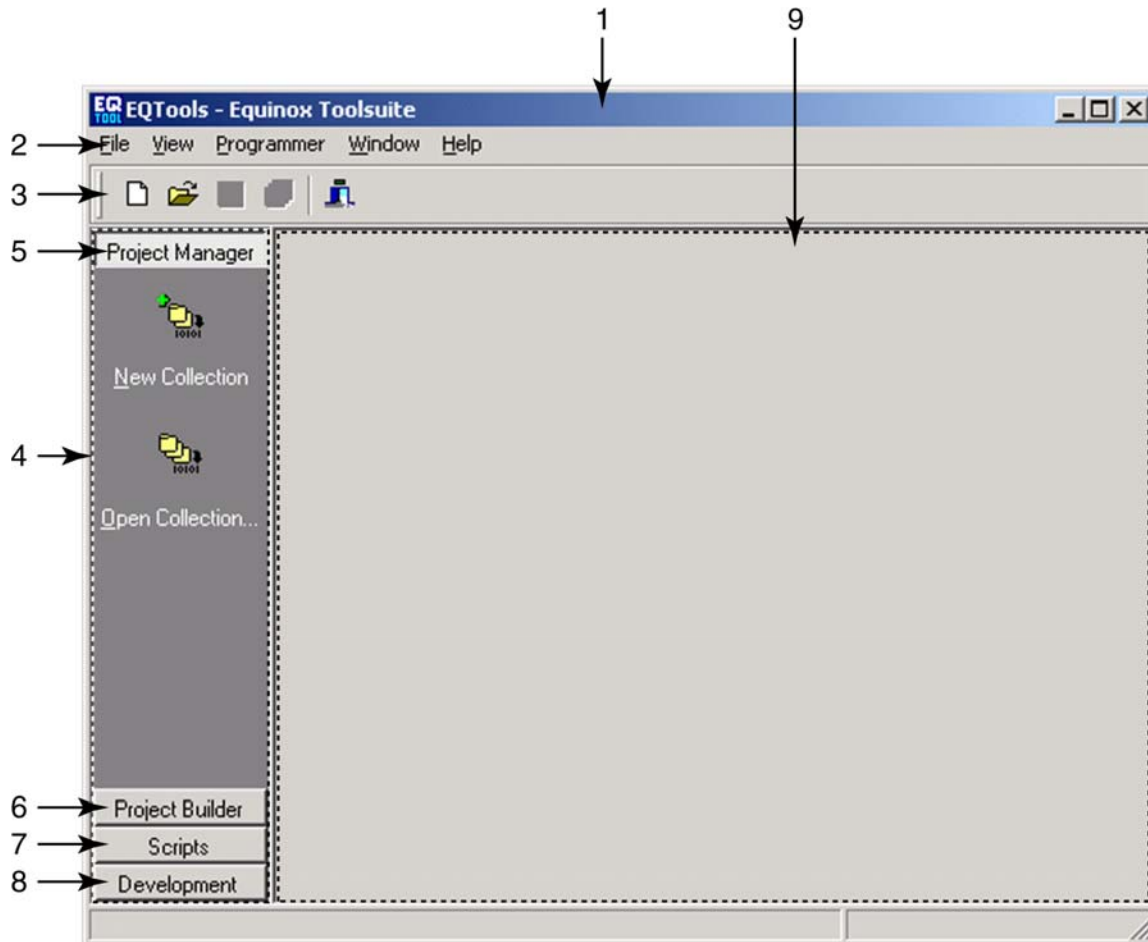


Fig. 4.2.1 EQTools – Main start window

EQTools defaults are as follows:

- Project Manager tab is selected
- No previously opened files are opened

Fig. 4.2.2 Key to EQTools – Main start window (fig. 4.2.1)

Key No.	Icon Name / menu position	Functional description
1	EQTools Title Bar	This is the main Title Bar of the EQTools application.
2	EQTools Menu bar	The place where all the main menu items for EQTools are displayed. To select a menu item, simply left-click the mouse on the required item and then select further items from the drop-down menu.

		Refer to section 4.4 for more information about the different tool bar items.
3	EQTools Toolbar(s)	This tool bar displays icons to control the various functions within EQTools. The 'Standard Toolbar' (left-hand 5 icons) will be displayed throughout all windows in EQTools. Further toolbars will be displayed to the right of this toolbar depending on what type of window is currently active.
4	EQTools – Utility Selection Pane	<p>This 'Selection Pane' is used to select one of the main EQTools utilities from the list of tabs (Project Manager, Project Builder, Scripts & Development)</p> <p>To select a particular utility, simply click on the required grey tab eg. for creating a new project, click on the <Project Builder> tab and then select <New Project> from the list displayed.</p>
5	Project Manager	<p>This tab is used for opening, creating and editing Project Collections. Simply click the grey tab header to display this tab and select <New Collection> or <Open Collection...>.</p> <p>Please refer to section 4.3 for detailed information.</p>
6	Project Builder	<p>This tab is used for opening, creating and editing single Projects. Simply click the grey tab header to display this tab and select <New Project> or <Open Project>.</p> <p>Please refer to section 4.3 for detailed information.</p>
7	Script Builder	<p>This tab is used for opening, creating and editing Script Wizard files and Script files. Simply click the grey tab header to display this tab and select <New Script Wizard> or <Open Script Wizard> or <New Script File> or <Open Script File...>.</p> <p>Please refer to section 4.3 for detailed information.</p>
8	Equinox Development Suite	This tab is used for testing projects in EDS. Simply click the grey tab header to display this tab and select <New Development Project> or <Open Development Project>.

		Please refer to section 4.3 for detailed information.
9	EQTools – Desktop	This is the main EQTools desktop where all EQTools windows will be displayed.

4.3 Overview of EQTools Components

4.3.1 Overview

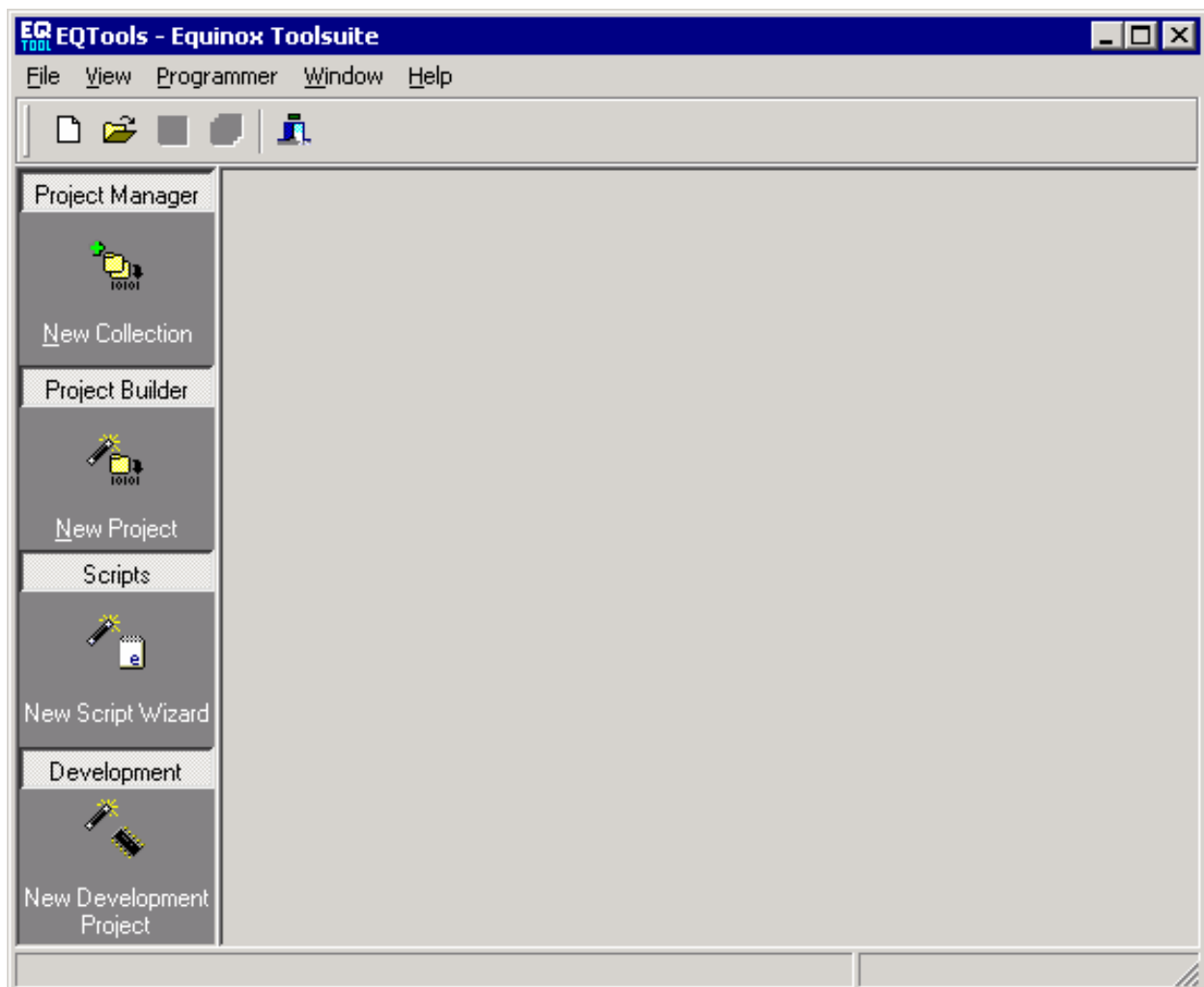
EQTools is an Integrated Development Environment (IDE) which comprises various utilities including the Project Builder, Project Manager, Project Upload Wizard, Script Builder and Script Debugger. An overview of the function of each utility is given in the table below. Please refer to section 4.3.2.

Utility	Description
<i>Project Builder</i>	<p>This utility allows you to create a 'Programming Project' which describes contains all the relevant information about how to program a target device on a specific target system.</p> <p>File types:</p> <ul style="list-style-type: none">• Programming Project (source file) - *.ppm• Programming Project (compiled file) - *.prj
<i>Project Manager</i>	<p>This utility allows you to create a 'Project Collection' into which you can add your 'Programming Project(s)'. The Project Collection can then be uploaded to the target programmer using the Upload Wizard.</p> <p>File types:</p> <ul style="list-style-type: none">• Project collection - *.ppc• Programming Project (compiled file) - *.prj
<i>Development Suite (EDS)</i>	<p>Equinox Development Suite (EDS) is a powerful development environment for debugging programming issues with the target device. EDS allows you to read/write the target device FLASH / EEPROM / FUSE and LOCK bits under PC control without the requirement to upload a 'Programming Project' to the programmer.</p> <p>File types:</p> <ul style="list-style-type: none">• Equinox Development Suite file - *.eds• Project collection - *.ppc• Programming Project (compiled file) - *.prj
<i>Upload Wizard</i>	<p>The Upload Wizard is used to upload the 'Programming Projects' contained within a 'Project Collection' to the 'FLASH Memory Store' on the target programmer. These projects are stored in non-volatile FLASH memory and so will be retained even if the programmer power is switched OFF and then ON again.</p> <p>File types:</p> <ul style="list-style-type: none">• Project collection - *.ppc

Script Builder	This utility is used to create 'Programming Scripts' which can be executed within the ISP-PRO application.
Script Debugger	This utility is used to debug / test 'Programming Scripts' which have been created with Script Builder before running them with ISP-PRO.

4.3.2 Launching an EQTools utility

To select the required utility, simply click once on the relevant tab in the left-hand pane of the EQTools window and either open or create a new file using the buttons <New ...> / <Open ...> or <File><New> / <File><Open> from the menu.



4.4 EQTools Toolbar Overview

4.4.1 Overview

EQTools features a number of 'Toolbars' which are used to control the different features / functions within the application. The 'Standard' toolbar is displayed throughout all windows in EQTools. Other toolbars will be displayed to the right of the 'Standard Toolbar' depending on the active window eg. Project Builder, Project Manager, Script Builder etc.

Fig 4.4.1.1 – Standard + Project Manager Toolbar example








4.4.2 Standard Toolbar

The icons detailed in the table below have the same function / meaning throughout EQTools and are displayed for all EQTools windows including Project Builder, Project Manager and Script Builder.

Fig 4.4.2.1 – Standard Toolbar



fig. 4.4.2.2 Standard Toolbar – Icon Functionality

Icon	Icon Name / menu position	Functional description
	New <File><New>	Creates a new file Opens a dialog allowing you to choose the file type and the appropriate wizard.
	Open <File><Open>	Opens a file Opens the standard Windows File Open dialog. This allows you to browse and select a file to be opened.
	Save <File><Save>	Saves the currently active file The currently active (selected) window is saved.
	Save all <File><Save All>	Saves all currently opened files All currently opened windows are saved.
	Exit <File><Exit>	Exits EQTools You will be prompted to save any unsaved files before quitting.

4.4.3 Project Builder Toolbar

The icons detailed in the table below are displayed whenever a Project Manager window is active. i.e. when a Project Builder Source File (*.ppm) is being edited.

Fig. 4.4.3.1 Project Builder – Example Screenshot

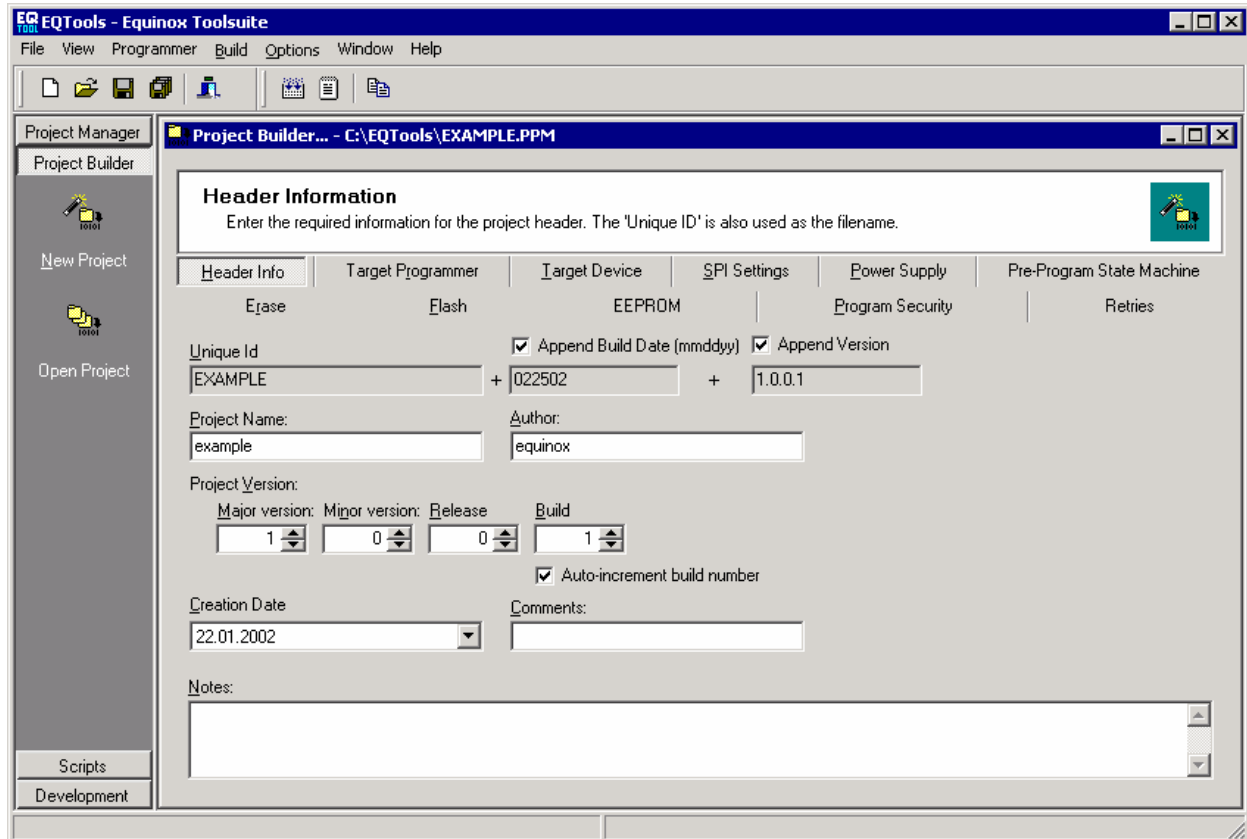


fig 4.4.3.2 Project Builder – Toolbar

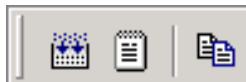





fig. 4.4.3.3 Project Builder – Toolbar Icon Functionality

Icon	Icon Name / menu position	Functional description
	Compile / Build Project N/A	Compile / build the active project Compiles a Project Builder Source File (*.ppm) into a Project Object File (*.prj). The *.prj file must then be added to a project Collection (*.ppc) before uploading it to a programmer.
	Export Project list <Edit><Export Project List>	Export Project list to text file Creates and opens a text file detailing the projects contained within the currently active Project

		Collection.
	Copy Unique ID <Edit><Test Project in EDS>	Copies Project Unique ID to clipboard Copies the 'Unique ID' (32 character unique name) of the currently selected project to the clipboard. This 'Unique ID' can then be pasted into the relevant field in eg. the Script Builder or Remote Application.

4.4.4 Project Manager Toolbar

The icons detailed in the table below are displayed whenever a Project Manager window is active. i.e. when a Project Collection File (*.ppc) is being edited.

fig. 4.4.4.1 Project Manager – Example Screenshot

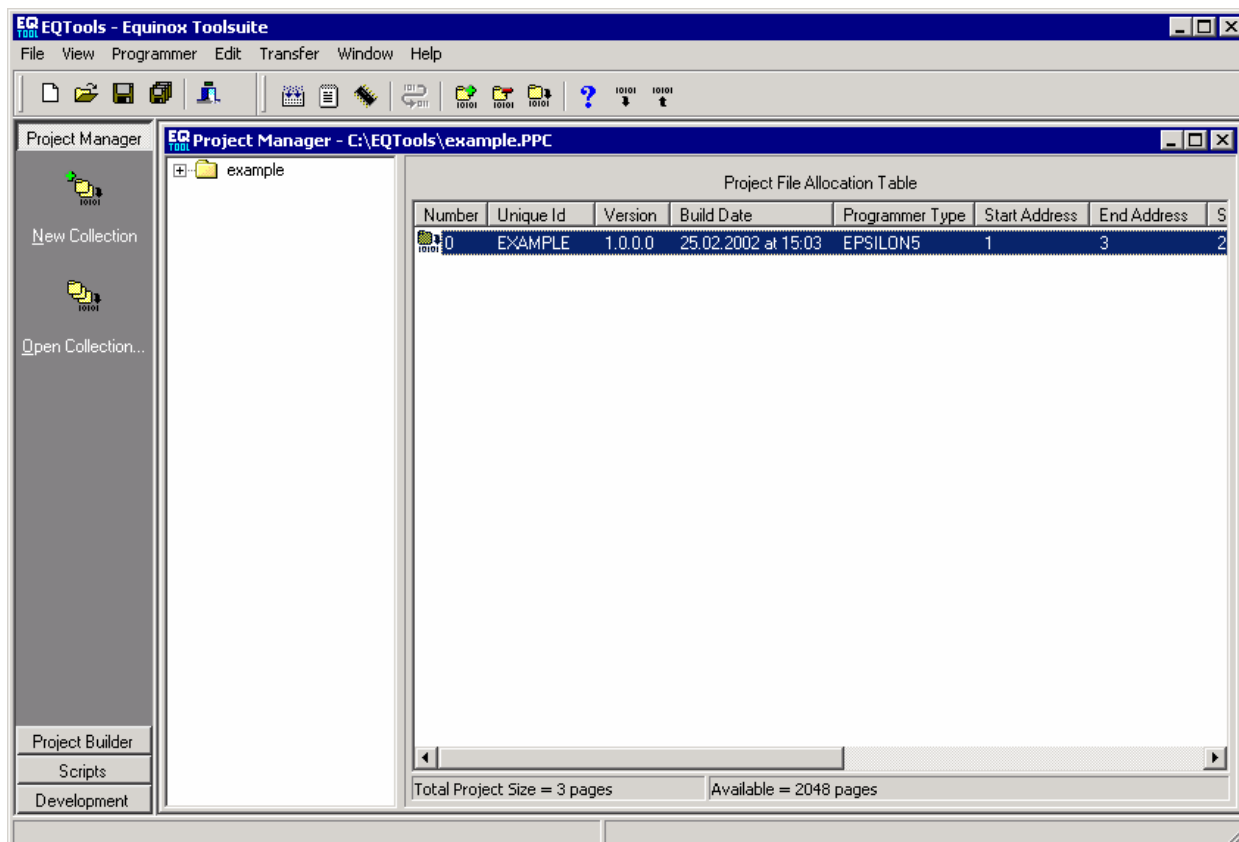












fig. 4.4.4.2 Project Manager - Toolbar



fig. 4.4.4.3 Project Manager - Toolbar Icon Functionality

Icon	Icon Name / menu position	Functional description
	Re-Build and Save Collection N/A	Re-builds active Project Collection and then saves it to disk Re-builds the currently active Project Collection and then saves the file (*.ppc) to disk. You will be prompted to update any component projects which have been updated since the last build of the collection.
	Export Project list	Export Project list to text file

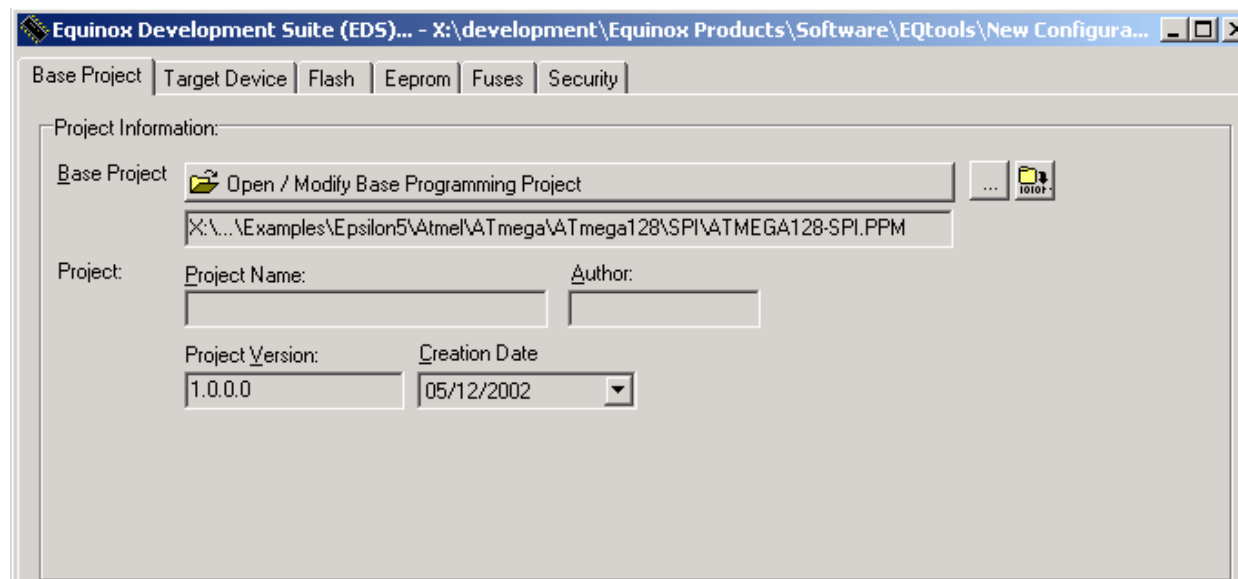
	<Edit><Export Project List>	Creates and opens a text file detailing the projects contained within the currently active Project Collection.
	Test Project in EDS <Edit><Test Project in EDS>	Test Project in Equinox Development Suite (EDS) Opens the selected project in Equinox Development Suite (EDS).
	Chained Project N/A	Chain two projects together Chains the currently selected project to the next project down in the list. This will cause the second project to be started immediately after the first one has been executed. Click this button again to unchain them.
	Add project to collection (Ctrl+A) <Edit><Add project...>	Add project to active Project Collection Opens a file open dialog box allowing you to select a Project Object file (*.prj) to add to the currently active Project Collection. → Once selected, this *.prj is added to the list of projects in the Project Allocation Table.
	Remove selected project from collection (Ctrl+D) <Edit><Remove project>	Removes selected project from active Project collection → The Project Allocation Table is updated automatically.
	Edit project <Edit><Edit project>	Edit Project Builder file (*.ppm) Opens the *.ppm file for the selected project in Project Builder. You can also double click on the project name in the Project Manager window to edit the project.
	Preview attached programmer(s) <Transfer><Preview>	Preview Project Allocation Table from attached programmer(s) Interrogates the attached programmer(s) and displays the Programming Projects which are resident in each programmer. Project Allocation Table for these programmers in the form of a new untitled Project Collection. This can be saved to disk if required.
	Download project data from programmer(s) (Ctrl+T) <Transfer><Transfer projects><Download>	Downloads Project Collection from a connected programmer to EQTools Interrogates the attached programmer(s) and displays the Programming Projects which are resident in each programmer in the form of a new untitled Project Collection. This can be saved to disk if required as a

		<p>*.ppc file.</p> <p>Note: The *.ppc does NOT contain the original Project Source Files (*.ppm).</p>
	Move up	Moves the selected project one place up the list of projects.
	Move down	Moves the selected project one place down the list of projects.
	Upload project data to programmer(s) (Ctrl+U) <Transfer><Transfer projects><Upload/Verify>	<p>Uploads the active project collection to the attached programmer(s).</p> <p>The projects (*.prj) contained in the active Project Collection are transferred to the connected programmer FLASH Memory Store.</p>

4.4.5 Equinox Development Suite (EDS) Toolbar

The Equinox Development Suite (EDS) does not need any additional icons so its toolbar is empty. The various features of EDS can be accessed by clicking on the relevant tab as shown in the picture below.

fig. 4.4.5.1 Equinox Development Suite – Example Screenshot



4.5 Re-opening files

4.5.1 Overview

EQTools maintains a history of which files have been opened in previous sessions of the program. When you launch EQTools, the application will not open any previously opened files automatically. It is possible to open previously opened files manually or to automatically re-open all files which were open when the last session of EQTools was closed. Please see the next two sections for further details.

4.5.2 Re-opening a previously opened file

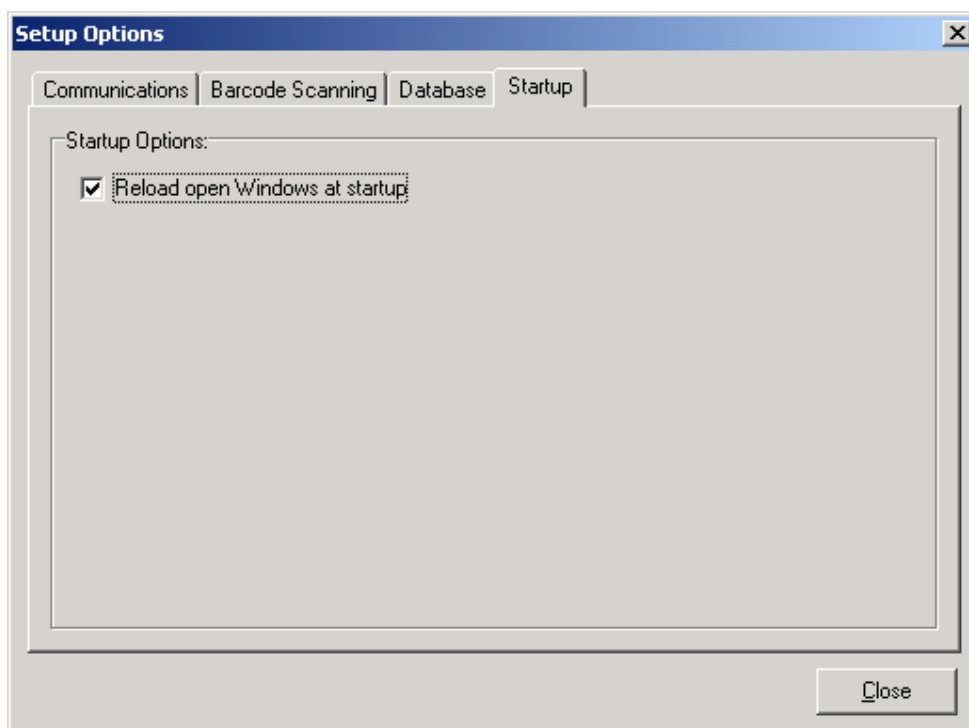
To re-open a file which has been opened/saved in a previous session of EQTools:

Select <File><Re-open> and then select the file from the drop-down file history list displayed.
→ The relevant file will be automatically opened.

4.5.3 Re-opening all files from last session of EQTools

If you wish to automatically re-open all files which were open when EQTools was last closed down:

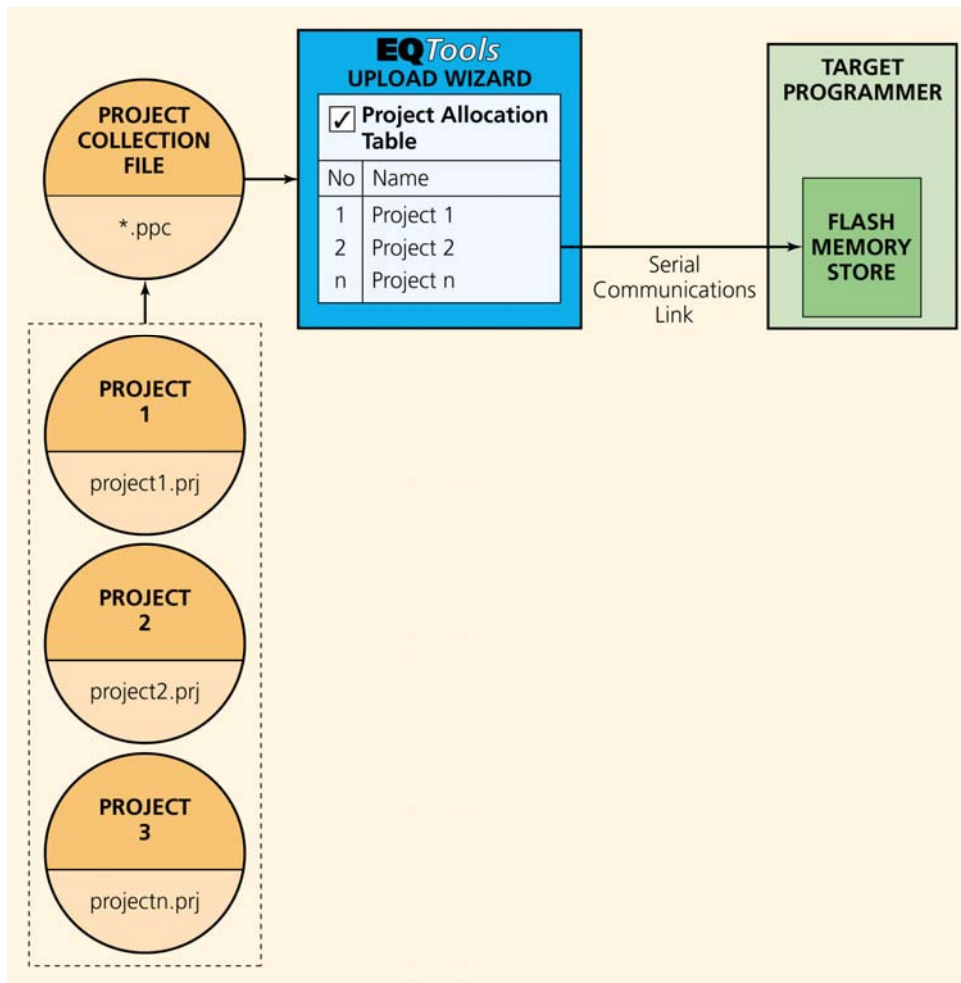
1. Select <Programmer><Communications and scripting setup><Startup>
→ The <Startup> tab is displayed.
2. Tick the box 'Reload open Windows at startup'



4.6 Uploading a Project Collection to a programmer

4.6.1 Overview

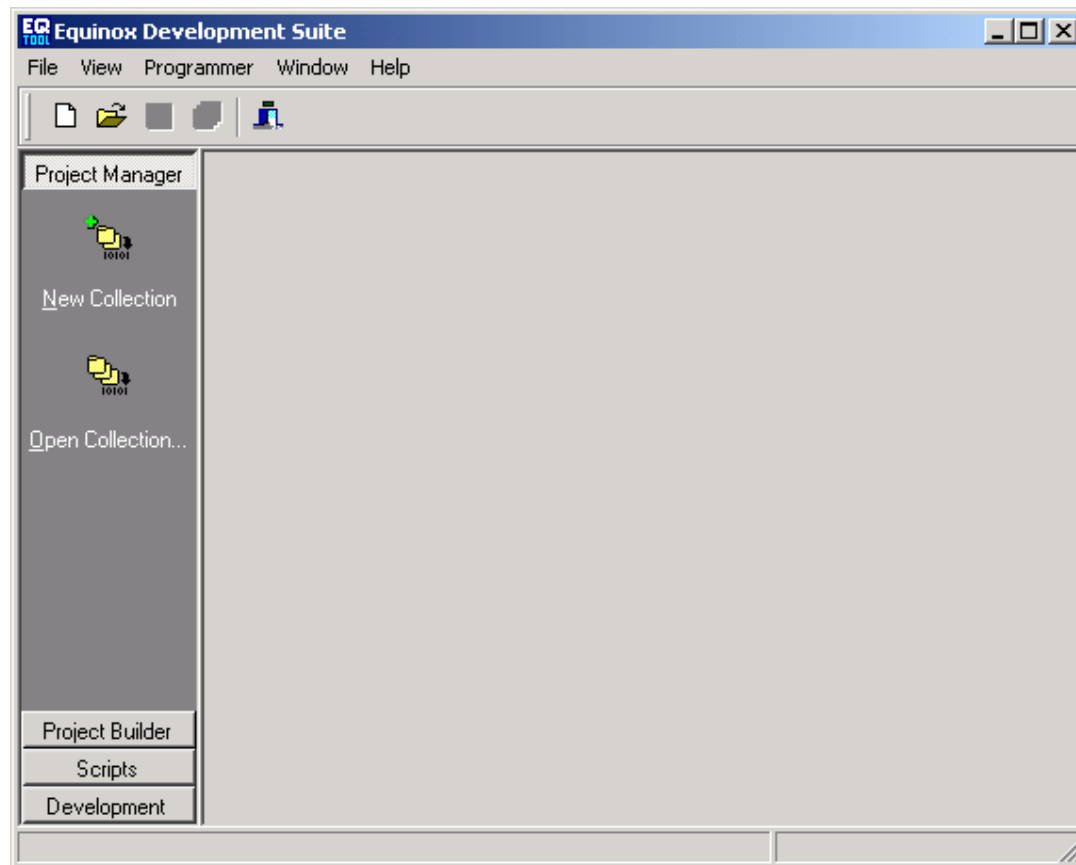
The instructions in this section detail how to upload a Project Collection to the programmer 'FLASH Memory Store'. This is shown diagrammatically in the figure below. The Project Collection file (*.ppc) contains all the individual Project Files (*.prj).



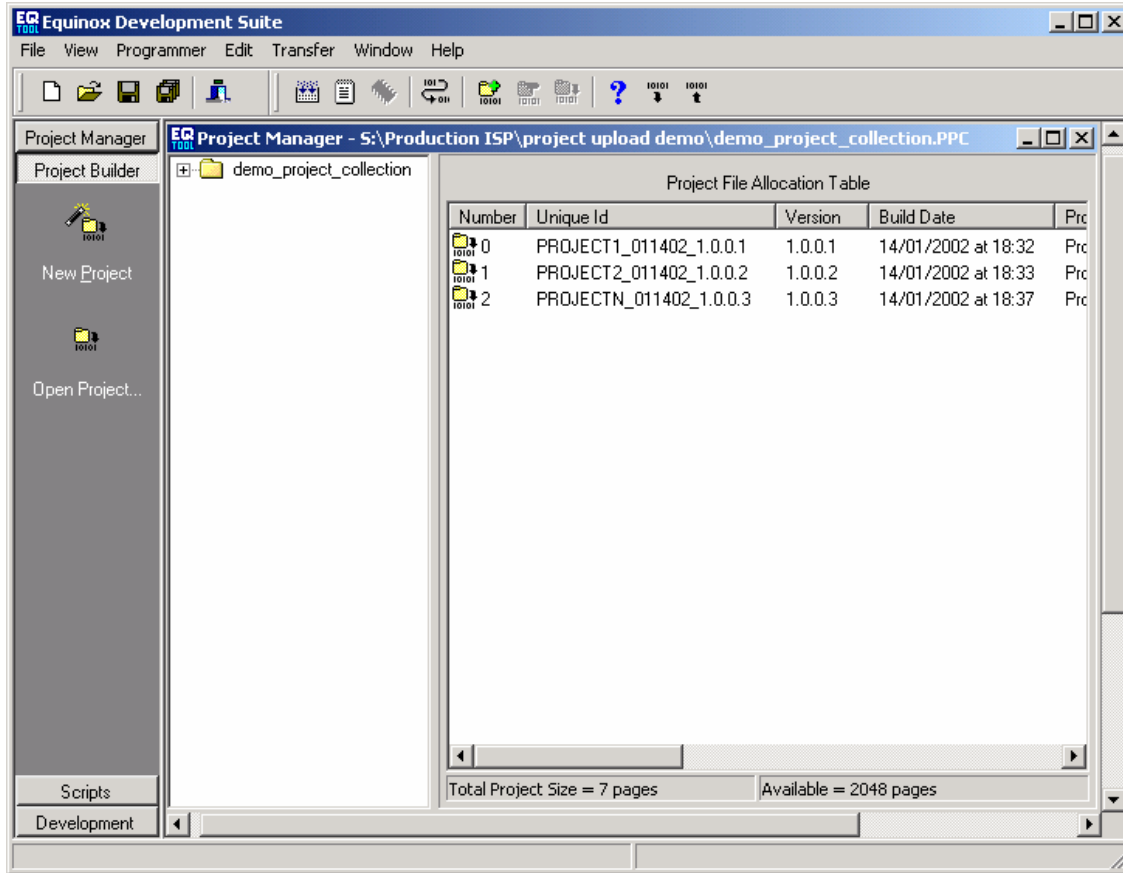
Please note: The EPSILON5 programmer is restricted to ONE project or TWO 'Chained Projects' resident in the programmer 'FLASH Memory Store' at any point in time.

4.6.2 Procedure

1. Launch EQTools
2. Select the <Project Manager> tab from the pane on the left-hand side of the screen.



3. Select <Open Collection>
4. Browse to the Project Collection file (*.ppc) which contains your projects.
→ The Project Manager Window should now open showing the details of all the projects contained in the Project Collection



5. Launch the Project Upload Wizard by any of the following methods

Click the Upload Project(s)  icon

Select a particular project from the Project Collection and then click the <Upload Selected Project> button

Click the <Upload all projects> to upload all projects in the Project Collection to the programmer.

→ The Project Upload Wizard will launch.

6. Follow the on-screen instructions to upload the selected project(s) to the attached programmer(s).

5.0 Equinox Development Suite (EDS)

5.1 Overview

The Equinox Development Suite (EDS) is a powerful tool to create and test your 'Programming Projects'. It can be used to read and write the target device FLASH / EEPROM memory areas and to enable or disable the fuses and lock bits. You do not need to upload a project to the programmer when working with EDS. Nevertheless, the settings of the compiled project will be used in this EDS session. This is quite useful for example when testing pre-programming state settings, since they are also applied in EDS.

5.2 Features

The basic features of EDS are listed below:

- Write / verify a block of data in the FLASH or EEPROM area of the Target Device
- Read a block of data from the FLASH or EEPROM area of a Target Device to the relevant EDS buffer.
- Read / Write the Target Device Configuration Fuses
- Read / Write the Target Device Security Settings / Fuses
- Perform CRC Checksum of both the Target Device and the EDS Buffer

5.3 Launching Equinox Development Suite (EDS)

The Equinox Development Suite or EDS can be used to create a new 'Test Project' from scratch or you can test an existing Programming Project which is in a Project Collection.

5.3.1 To make a new Development Project using the EDS Wizard

It is very straightforward to set up EQTools to program a device in a Development Environment. The new EDS Wizard allows you to simply select the Target Programmer, Device and Power Supply Characteristics and then automatically launches Equinox Development Suite (EDS). The wizard sets all other settings to default values based on the chosen device and target voltage.

To test a Target Device device using Development Mode (EDS):

- Launch EQTools
 - From the 'Welcome Screen' select '**Create a new Development Project**'.
- Or
- From the left-hand pane of the EQTools Window, select the Development tab and then select '**Create a new EDS Project**'.

→ The EDS Wizard is now displayed.

This wizard allows you to quickly set up the target programmer, device and power supply. You can also optionally specify the FLASH and EEPROM files here if required.

The steps in the EDS - Development Wizard are as follows:

- Select Programmer
- Select Target Device to program
- Select Target Voltage (and current if supported)
- Select FLASH Area File (optional)
- Select EEPROM Area File (optional)

To advance through the Wizard, click <Next>.

You can specify the files to load into the FLASH and EEPROM areas in the Wizard or you can leave these tabs blank and manually load any files once you are running EDS.

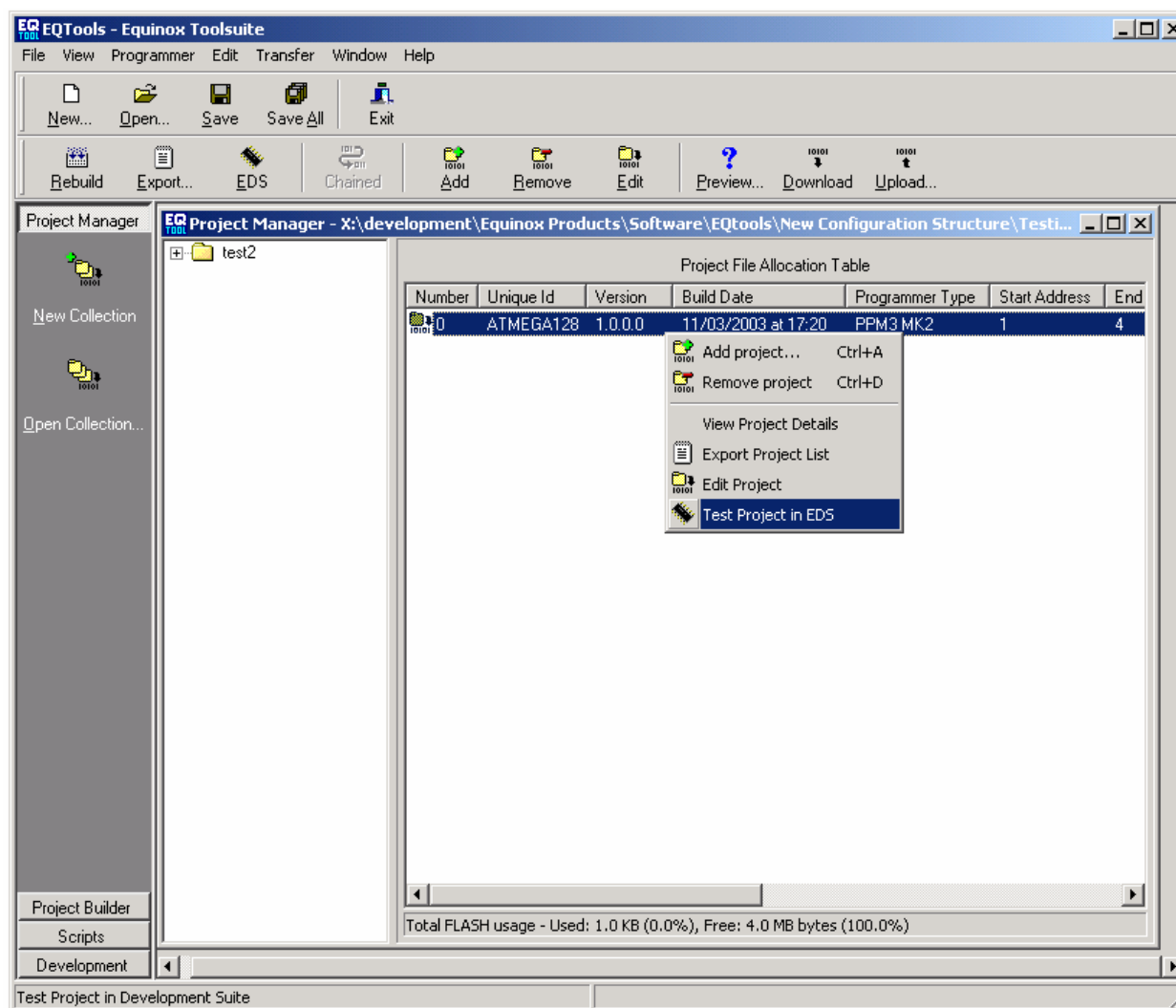
At the end of the Wizard:

- Click the <Test> button
- Enter a suitable file name for your EDS Project e.g. ATtiny26.ppm (You can just use the default name EDSTEST.ppm if you are not planning on using this project again.)

--> EDS window launches with your specified project loaded.

5.3.2 To test a pre-compiled Programming Project in EDS

- i. Open your Project Collection which contains the Programming Project(s) which you wish to test in EDS. (Select <Open Collection> from the left-hand pane and then browse to your Project Collection.)
- ii. Locate the Programming Project you wish to test in the Project Manager window
- iii. Click once on the name of the project → project is highlighted in blue.
- iv. Click the <Test in EDS> button or click the EDS icon or right-click the mouse and select <Test in EDS> from the drop-down menu.

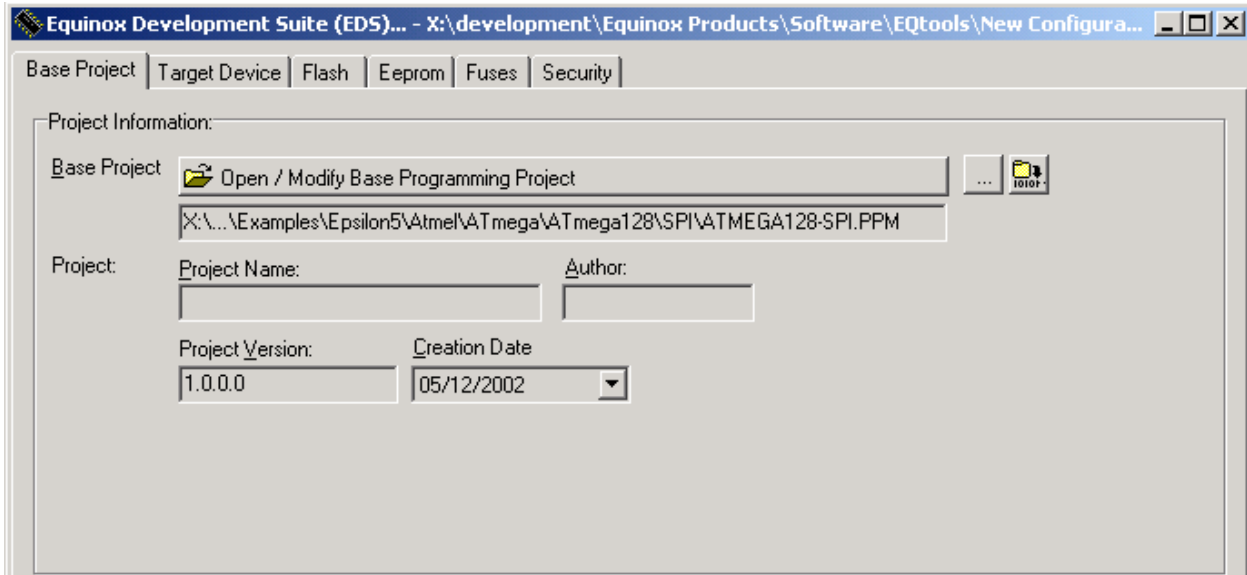


→ EDS window will now launch and will default to the <Base Project> tab – see next section.



5.4 <Base Project> tab

5.4.1 Overview

The <Base Project> tab displays the details of the Programming Project which is currently loaded into EDS as shown in the screenshot below.



The following options are available on this tab:

- Click <Open/Modify Base Programming Project> to re-open the current loaded Project Source File (*.ppm)
- Click on  to select a different Project Source File (*.ppm) to load into EDS
- Click on  to create a new project.

If you used the 'EDS Wizard' to create your EDS Project, then the Base Project will show the name you chose for the file at the end of the Wizard. E.g. If you chose Test.eds, the Base Project will be automatically called 'Test.ppm'.

5.4.2 Editing programming settings in EDS

EDS actually creates a Programming Project (*.ppmm) in the background which stores all the settings you have made within the EDS session. It is possible to open this project and use it as a Standalone Project if required.

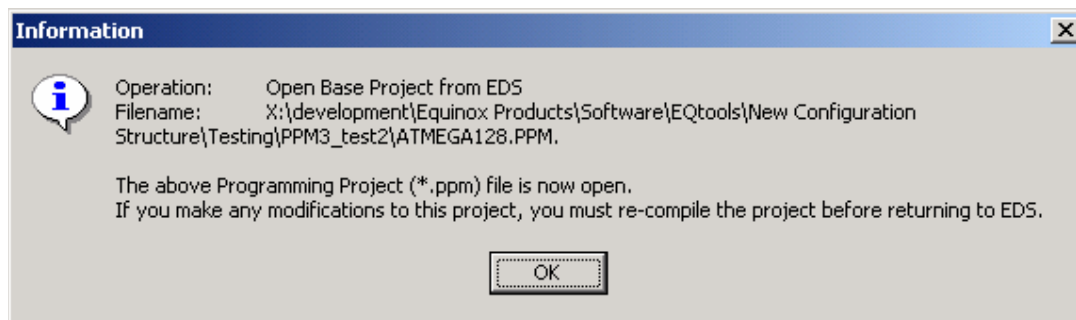
To edit a setting in the Programming Project:

1. Click <Open/Modify Base Programming Project> to re-open the current loaded Project Source File (*.ppm)

→ The EDS window is automatically closed.

→ The specified 'Programming Project' source file (*.ppm) is opened in the Project Builder window.

→ The dialog below is displayed which reminds you to re-compile the project after making any changes to it. Click <OK> to start editing the project.




2. Make your changes to the 'Programming Project' source file (*.ppm)

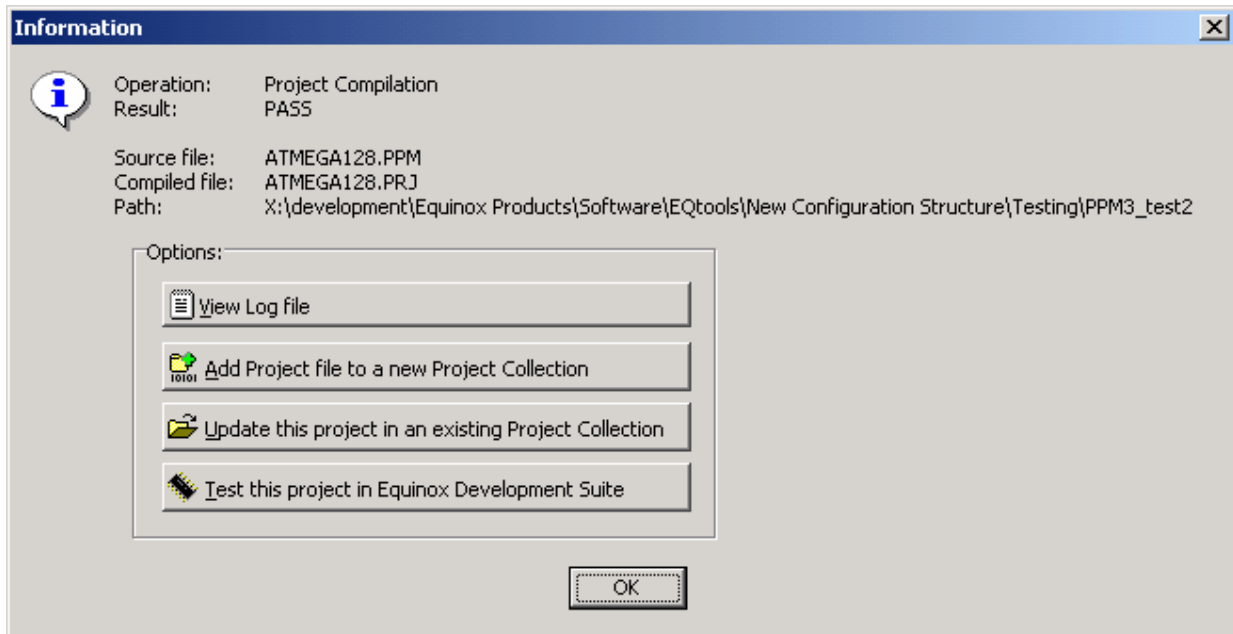
The specified 'Programming Project' source file (*.ppm) is opened in the Project Builder window. You can now change the Target Device / SPI frequency / Pre-programming State Machine timings etc.

3. Re-compile the 'Programming Project' source file (*.ppm)

Once you have completed your changes to the Programming Project, click the <Compile>

icon  on the top icon bar.

→ If the project compiles OK, the dialog box below is displayed.



Click the **<Test this project in Equinox Development Suite>** to load the project (*.ppm) into EDS.

→ EDS window now launches and loads the new settings from the specified programming 'Programming Project' *.ppm file.

4. Re-test your Programming Project in EDS

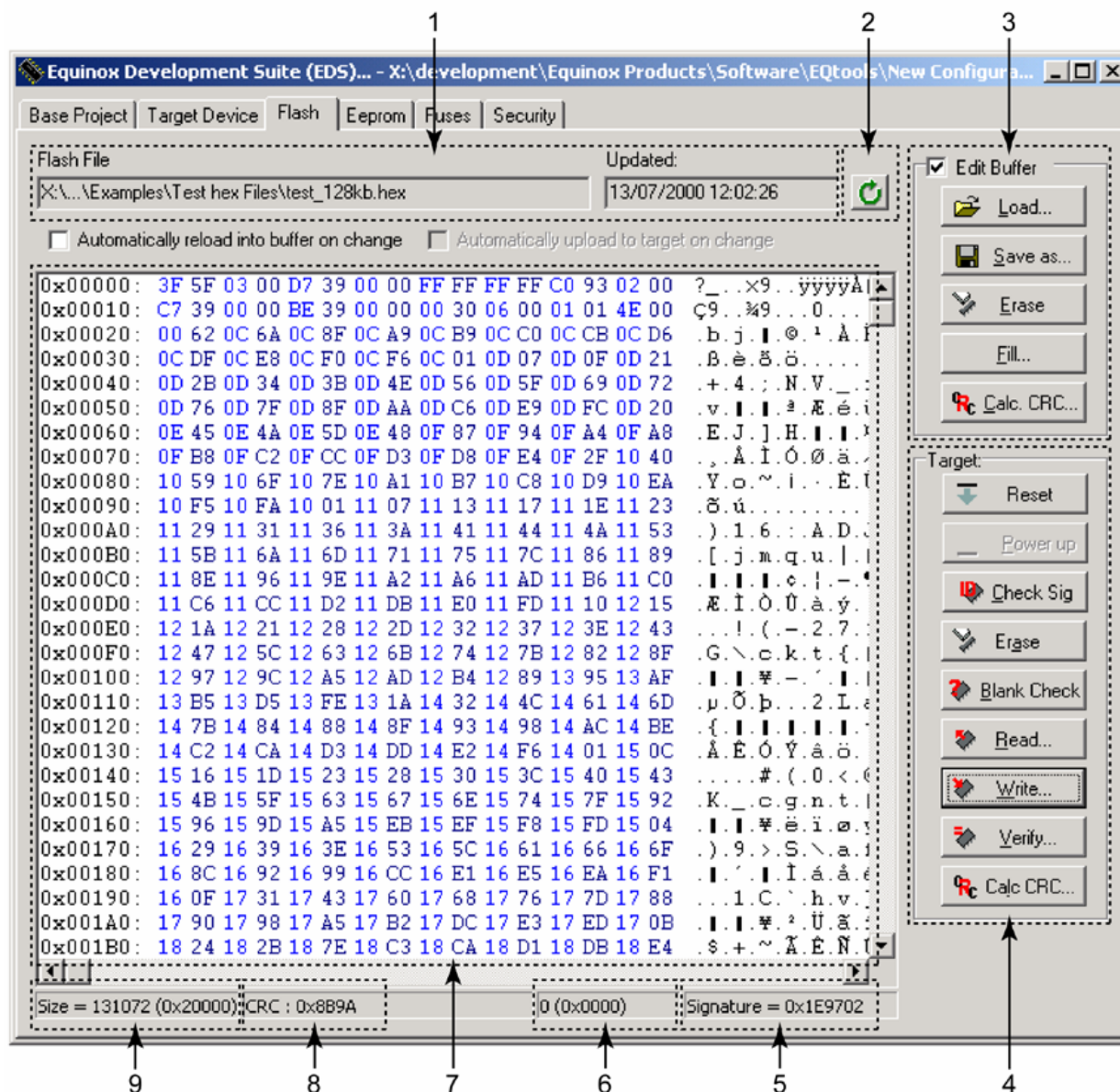
Click on the relevant tab in the EDS window to select the required function(s) to test.

5.5 <Flash / EEPROM> tabs


5.5.1 Overview

The FLASH and EEPROM tabs are identical in layout. The FLASH tab supports programming operations on the FLASH area of the Target Device. The EEPROM tab supports programming operations on the EEPROM area of the Target Device.

The layout of the EDS – <FLASH> tab is shown in the screenshot below:




FLASH / EEPROM tab – functional description

#		
1	File name and properties	<p>This specifies the 'File Name / path' and 'Last updated' date of the file to be loaded into the Buffer Window.</p> <ul style="list-style-type: none"> • By default, this will point to the file specified in the associated Programming Project (*.ppm). • To load a different file, click the <Load> button and browse to the required file.
2	Re-load file to buffer	<p>Clicking the  icon re-loads the specified Flash / EEPROM file into the Buffer Window. This will overwrite any information already in the Buffer Window.</p>
3	Buffer Control Buttons	<p>This group of buttons control operations on the EDS – Buffer Window.</p> <ul style="list-style-type: none"> • To use these buttons or to manually edit the Buffer Window, it is necessary to check the 'Edit Buffer' check box. • This extra step helps to avoid accidental modification of data in the buffer.
4	Target Control Buttons	<p>This group of buttons control operations on the actual Target Device.</p> <ul style="list-style-type: none"> • A programmer and suitable Target System must be connected when using these buttons.
5	Signature of Target Device	<p>This is the signature (Device ID) which is expected for the Target Device.</p>
6	Buffer Address	<p>This is the address of the currently selected location in the Buffer Window.</p>
7	Buffer Window	<p>The Buffer Window displays a hexadecimal and alphanumeric representation of the data which has either been loaded from file or read back from a Target Device.</p> <ul style="list-style-type: none"> • The bytes are grouped into rows of 16 bytes with the start address of each row displayed in the left-hand column. • The Hexadecimal representation of the 16 bytes is displayed in the middle column • The ASCII representation of the 16 bytes is displayed in the right-hand column.
8	CRC	<p>This is a CRC Checksum of the entire Buffer (ie. from address 0x00000 to the end address specified in the buffer.</p>
9	Size	<p>This is the physical address range in bytes of the entire Buffer.</p>

5.5.2 Buffer Window - Control Buttons

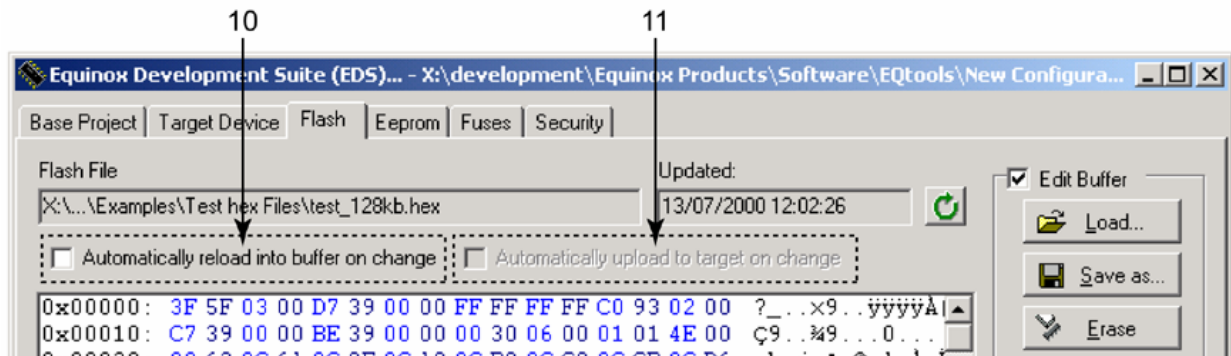
The Buffer Window within EDS is used as a virtual data store which allows data to be transferred from file → Buffer Window → Target Device and vice-versa.


The 'Buffer – Control Buttons' support operations on the data in the Buffer Window including loading a file into Buffer or saving the contents of the Buffer to file. These buttons do NOT initiate any actual programming operation of the Target Device.


#	Button	Description / Purpose
1	 Re-load File	<ul style="list-style-type: none"> Re-loads the specified HEX / Binary file into the Buffer Window.
2	Edit Buffer	<ul style="list-style-type: none"> Tick the box to enable manual editing of the Buffer Window. This will also allow the <Load>, <Erase> and <Fill> buttons to be selected.
3	<Load>	<ul style="list-style-type: none"> This button allows you to manually browse to select a file to be load into the Buffer Window. The path will default to the path set up in the Programming Project (if a file is specified).
4	<Save as>	<ul style="list-style-type: none"> This button allows you to save the contents of the Buffer Window to a file. <p>It is possible to specify the following properties for the file:</p> <ul style="list-style-type: none"> File type (HEX / Binary) File Start and End Address
5	<Erase>	<ul style="list-style-type: none"> This button erases the entire contents of the Buffer Window. All locations are set to the value 0xFF.
6	<Fill>	<ul style="list-style-type: none"> This button allows you to fill a specified area of the Buffer Window with a specified value.
7	Calc CRC	<ul style="list-style-type: none"> This button calculates the CRC Checksum of the specified range of the Buffer Window.

5.5.3 File loading into the Buffer Window

A valid FLASH or EEPROM file can be loaded into the Buffer Window using one of the methods below:



1. Click the press  icon
 - This will automatically load the specified FLASH or EEPROM file into the Buffer Window.
 - This will only work if a valid file path / name has been declared in the Programming Project (*.ppm) file.
 - If no file name / path is specified, either go back the Programming Project and specify a file or Click the Buffer <Load> button – see (2).
2. Click the Buffer <Load> button
 - Make sure the 'Edit Buffer' check box is ticked. This will allow the <Load> button to be selected and the Buffer Area to be edited.
 - Click the <Load> button → a file browser is displayed.
 - Browse to and select the required FLASH or EEPROM file.
→ The file name and path will now be displayed in the EDS window.

Clicking the press  icon at any time will re-load the specified file into the Buffer Window.

5.5.4 Automatically reloading a file on change

In a development environment it is often the case that the source code is being continuously changed and then re-compiled to produce a new Hex file. This file then needs to be programmed into the Target Device to test whether the source code changes had the desired effect. To cater for this requirement, EDS supports automatic reloading of the specified FLASH / EEPROM file into the buffer.

To automatically re-load the file into the buffer / device on change:

- Tick the 'Automatically reload into buffer on change' tick box.
- Keep the EDS Window open
- Re-compile your CODE source file using your assembler or compiler
→ The Hex / binary file date and time properties will be updated.
- Click anywhere on the EDS Window
→ The specified HEX / binary will now be automatically loaded into the Buffer Window.
- If you wish for EDS to also automatically program this file into the Target Device, tick the 'Automatically upload to target on change' check box.

5.5.5 Saving data from the Buffer to file

It is possible to save the contents of the Buffer Window to file. This is useful when the contents of a Target Device have been read into the Buffer Window and a permanent copy of this data is then required.

To save the contents of the Buffer Window to file:

- Click the <Save as> button → 'Save as' dialog is displayed.



- Enter the 'Start Address' in the buffer where data is to be saved from.
- Enter the number of bytes to be saved or a valid 'End Address'.
- Enter an offset address if required (HEX files only). This will place an offset to the Start Address so that when the file is re-loaded it will start at the 'Offset Address'.
- Click <OK> → File Browser is displayed.
- Select the required file type (default is Intel HEX)
- Type in a suitable file name and click <Save>
- The specified Buffer Window address range is now saved to file.

5.2.6 Editing data in the Buffer Window

To manually edit the HEX data in the Buffer Window:

- Tick the box to enable manual editing of the Buffer Window.
- This will also allow the <Load>, <Erase> and <Fill> buttons to be selected.
- Click on the location in the Buffer Window which you wish to edit. This can either be in the HEX middle column or in the ASCII right-hand column.
- Manually edit the value
- Remember – this data has only been changed in the Buffer Window. It has not been changed in the Target Device.

5.2.7 Erasing the contents of the Buffer Window

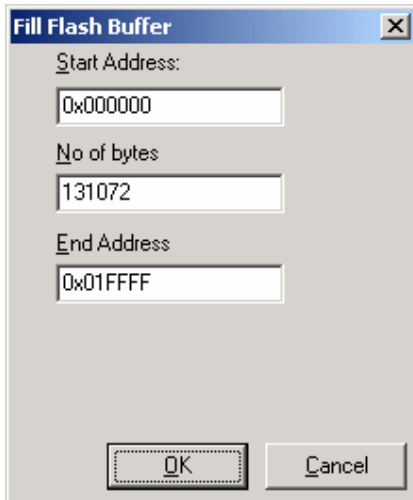
Clicking the Buffer <Erase> button will erase the contents of the Buffer Window setting all locations to the default erased value of 0xFF. This operation only erases the Buffer Window. It has no effect on the Target Device.

5.5.6 Filling the Buffer Window with a specified value

The <Fill> button allows you want to fill large blocks of the Buffer Window with the same value.

To fill a specified address range with a value:

- Click the <Fill> button



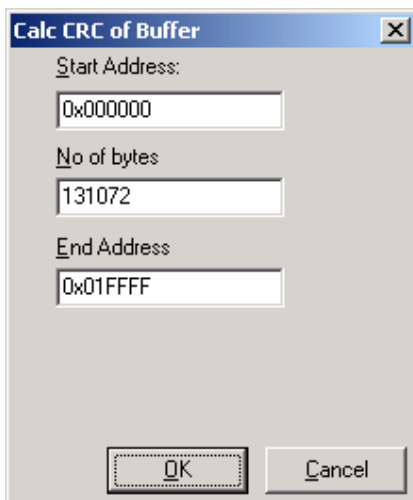
- Enter the required Start Address
- Enter either the number of bytes or the end address of the block to be filled.
- Enter the value to written to the block in the 'Fill with:' field
- Click <OK> → The specified block is filled with the specified value.

5.5.7 Calculating the CRC Checksum of the Buffer Window

The CRC Checksum of the entire Buffer Window is constantly displayed at the bottom of the Buffer Window.

If you wish to perform a CRC check on a specified address range of the buffer, please follow the instructions detailed below:

- Click the Buffer <Calc. CRC> button
→ The Calc CRC dialog is displayed.



- Enter the required Start Address
- Enter either the number of bytes or the end address of the block to be CRC checked.
- Click <OK> → The specified block is CRC checked and the calculated CRC value is then displayed.

5.5.6 Target Control Buttons

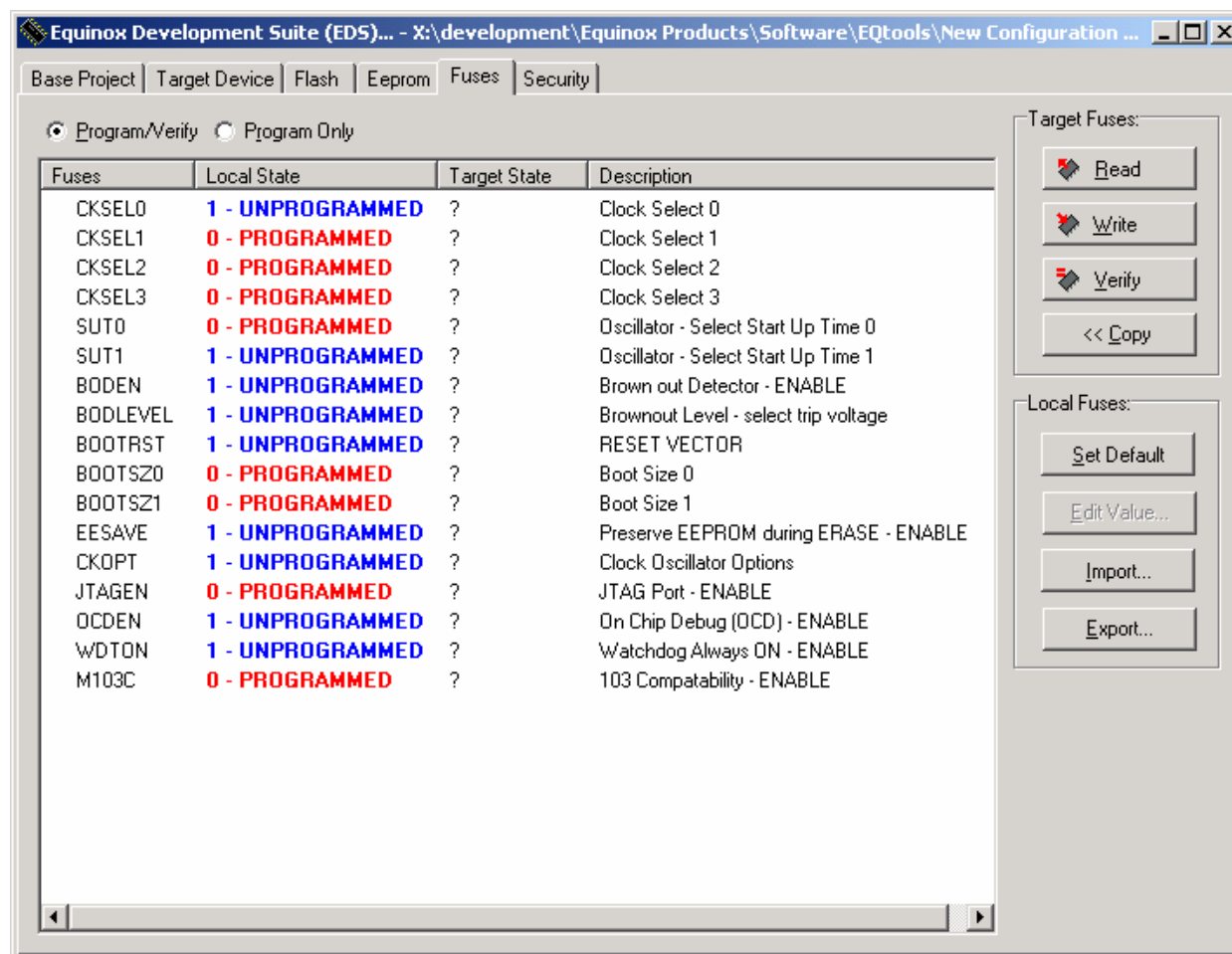
The Target Control Buttons control the actual programming operations of the Target Device.

#	Button	Description / Purpose
1	RESET	<ul style="list-style-type: none"> The RESET button resets the programmer. This will tri-state all the programmer I/O lines which should cause the Target System to run code.
2	Power Up	<ul style="list-style-type: none"> This button will signal the programmer to apply power to the Target System. This will only work if the programmer is capable of supplying controlled power. i.e. PPM3 MK2 programmer only.
3	Check Sig	<ul style="list-style-type: none"> This button checks that the Target Device Signature (ID) matches the signatures declared in the Programming Project.
4	Erase	<ul style="list-style-type: none"> This button issues a 'Chip Erase' command to the Target Device. For most devices this will erase the entire FLASH and EEPROM areas of the device to the default 0xFF value. For Atmel ATmega devices where the 'EESAVE' fuse has been ENABLED, the Erase operation will only erase the FLASH area, not the EEPROM.
5	Blank Check	<ul style="list-style-type: none"> This button performs a Blank Check of the Target Device.
6	Read	<ul style="list-style-type: none"> This button reads the contents of the Target Device to the EDS Buffer Window. This will only work properly if the Target Device Security has not been set.
7	Write	<ul style="list-style-type: none"> This button allows a specified address range from the buffer to be programmed / verified in the Target Device.
8	Verify	<ul style="list-style-type: none"> This button performs a Verify Only operation of the specified address range of the Target Device.
9	Calc. CRC	<ul style="list-style-type: none"> This button performs a CRC Checksum calculation over the specified address range of the Target Device.

5.6 <Fuses> tab

5.6.1 Overview

The EDS <Fuses> tab displays a list of any so-called 'Configuration Fuses' which are available for the currently selected Target Device. The function of each of these fuses depends on the Target Device and so the datasheet for the device should be studied carefully before attempting to modify the Fuse Settings.



Fuses

This is the actual name of the Fuse as specified on the datasheet of the Target Device.

PC Fuse State (Local State)

The PC Fuse state of the Fuse within EQTools – EDS can be altered in this column by simply double-clicking on the row of the relevant fuse.

- This will change the value of the Fuse in this window, but will NOT write this value to the Target Device.
- Writing the 'Local State' fuses to the Target Device is performed by clicking the <Write> button.
- The colour of the Fuse will change (Boolean Fuses only) to indicate when the fuse is ENABLED or DISABLED.

- Please note – the Local State Fuse column is updated on every re-launch of EDS to the default Fuse Settings found in the Device Library. If you wish EDS to remember your own custom Fuse Settings, you must export your Fuse Settings to a Fuse File (*.eff) and then import them back into EDS using the <Import> button on the fuse Tab.

Target State

This is the state of the Fuse in the actual Target Device.

- When the <Fuses> tab is initially displayed, the 'Target State' of all fuses will be displayed with a '?' as the state is currently unknown.
- To view the actual state of the Fuses in the Target Device, click the <Read> button.

Description

This is a simple description of the function of the Fuse.

A more detailed description can be obtained by studying the datasheet of the Target Device.

5.6.2 Fuse Types

EQTools currently supports two different Fuse Types as detailed in the table below:

#	Fuse Type	Description / Purpose
1	Boolean	<ul style="list-style-type: none"> • A Boolean Fuse can only have the value of '0' or '1'. • Boolean Fuses are typical of all Atmel AVR and AT89S devices where a '0' means the fuse is ENABLED and a '1' means the fuse is DISABLED.
2	HEX	<ul style="list-style-type: none"> • A HEX Fuse can have any value from 0x00 to 0xFF hex. • Boolean Fuses can be found on Atmel T89C51Rx2 microcontrollers and Philips P89C51Rx2 / 66x microcontrollers.

Each Fuse can also be tagged as a 'Read only', 'OTP' or 'Dangerous' Fuse as detailed in the table below/

#	Fuse Type	Description / Purpose
1	Read Only	<ul style="list-style-type: none"> • This type of Fuse can be read by the programmer but its value cannot be changed.
2	OTP	<ul style="list-style-type: none"> • One Time Programmable Fuses • These fuses can be programmed ONCE only. • Programming of an OTP fuse may stop the programmer from further communication with the Target Device. Please refer to the datasheet of the Target Device for further instructions.
3	Dangerous	<p>Warning !</p> <ul style="list-style-type: none"> • Any fuse which is marked as a 'Dangerous Fuse' could cause spurious behaviour of the Target Device if programmed. • For example, incorrect programming of the Atmel ATmega CKSEL fuses could render the Target Device dead to any further programming operation.

5.6.3 Target Fuses - buttons

The table below details the functions of the Target Fuses buttons. Each of these buttons performs an operation on the Fuses of the actual Target Device. Special care should therefore be taken before carrying out any operation as improper writing of the fuses could cause the Target Device to no longer communicate with the programmer.

#	Button	Description / Purpose
1	Read	<ul style="list-style-type: none">This button reads the value of each Fuse from the Target Device and then displays the value in the 'Target State' column.
2	Write	<ul style="list-style-type: none">This button writes the current 'Local State' of the each fuse to the Target Device.The Target Device fuses are then automatically read back and verified against the 'Local State' Fuse values.
3	Verify	<ul style="list-style-type: none">This button performs a Verify Only operation with the Target Device Fuses and the Fuse values declared in the 'Local State' column.The Target Fuses are also read back and listed in the 'Target State' column.
4	<<Copy	<ul style="list-style-type: none">This button copies the values of the 'Target State' column to the 'Local State' column.This allows the Fuse settings of a Target Device to be read out and then saved to a 'Fuse File' using the <Export> button.

! Warning!

It is possible to lose contact with the target device by setting certain combinations of `Fuse Settings.

For Atmel ATmega devices:

Please carefully check settings for the `CLKSEL' and `RSTDISBL' fuses BEFORE programming them into a target device. Once these fuses are incorrectly set, it may be necessary to manually apply an oscillator to the target device in order to re-program it again.

For Philips P89C51Rx2 and 66x devices:

Make sure that the 'BVA' fuse is not unintentionally written to a value other than 0xFC as the device will then no longer communicate with an ISP programmer.

5.6.4 Local Fuses - buttons

The table below details the functions of the Local Fuses buttons.

#	Button	Description / Purpose
1	Set Default	<ul style="list-style-type: none"> This button loads the default values for each Fuse as defined in the Target Device manufacturer's datasheet. These settings usually represent the state of the fuses of a virgin device (i.e. not programmer before).
2	Edit Value	<ul style="list-style-type: none"> This button allows you to edit the value of a Fuse. It is also possible to edit a Fuse by simply double-clicking on the Fuse in the list.
3	Import	<ul style="list-style-type: none"> This button allows a pre-specified list of Fuse Settings to be loaded into the 'Local State' column from an Equinox Fuse File (*.eff). The Fuse File must have already been created by exporting a Fuse File either in this window or in the Project Manager – <Fuse Settings> tab.
4	Export	<ul style="list-style-type: none"> This button exports the Fuse settings as declared in the 'Local State' column to an Equinox Fuse File (*.eff). This Fuse File can then be imported back into any Programming Project in the Project Manager – <Fuse Settings> tab. This is useful to copy the known fuse settings from a device to a file.

5.6.5 Exporting / Importing Fuse Settings to / from File

One of the new power features of EQTools is the ability to export the Fuse Settings for a Programming Project to a Fuse File (*.eff). This allows the settings for all the fuses to be contained in one Fuse File which can then be imported into any of the Fuse tabs in Project Manager or in the <Fuses> tab in EDS.

Exporting the Fuse Settings to a Fuse File

To export the settings of the 'Local Fuses' column to a fuse File:

- Select the EDS <Fuses> Tab
- Set up the 'Local Fuses' to the correct values for your Target Device.
- Click the <Export> button → a file browser is displayed.
- Enter a suitable name for your Fuse File eg. **project_fuses.eff**
- Click <Save> → The 'Local Fuses' column is transferred to your specified Fuse File.

Copying the Fuses from a Target Device

To copy the Fuses from the Target Device to a Fuse File:

- Select the EDS <Fuses> Tab
- Click the <Read> button
→ the Fuses are read from the Target Device and are then displayed in the Target State column.
- Click the <<Copy button
→ the Fuse settings read from the Target Device are copied into the 'Local State' fuse column.
- Click the <Export> button → a file browser is displayed.
- Enter a suitable name for your Fuse File eg. **project_fuses.eff**
- Click <Save> → The 'Local Fuses' column is transferred to your specified Fuse File.

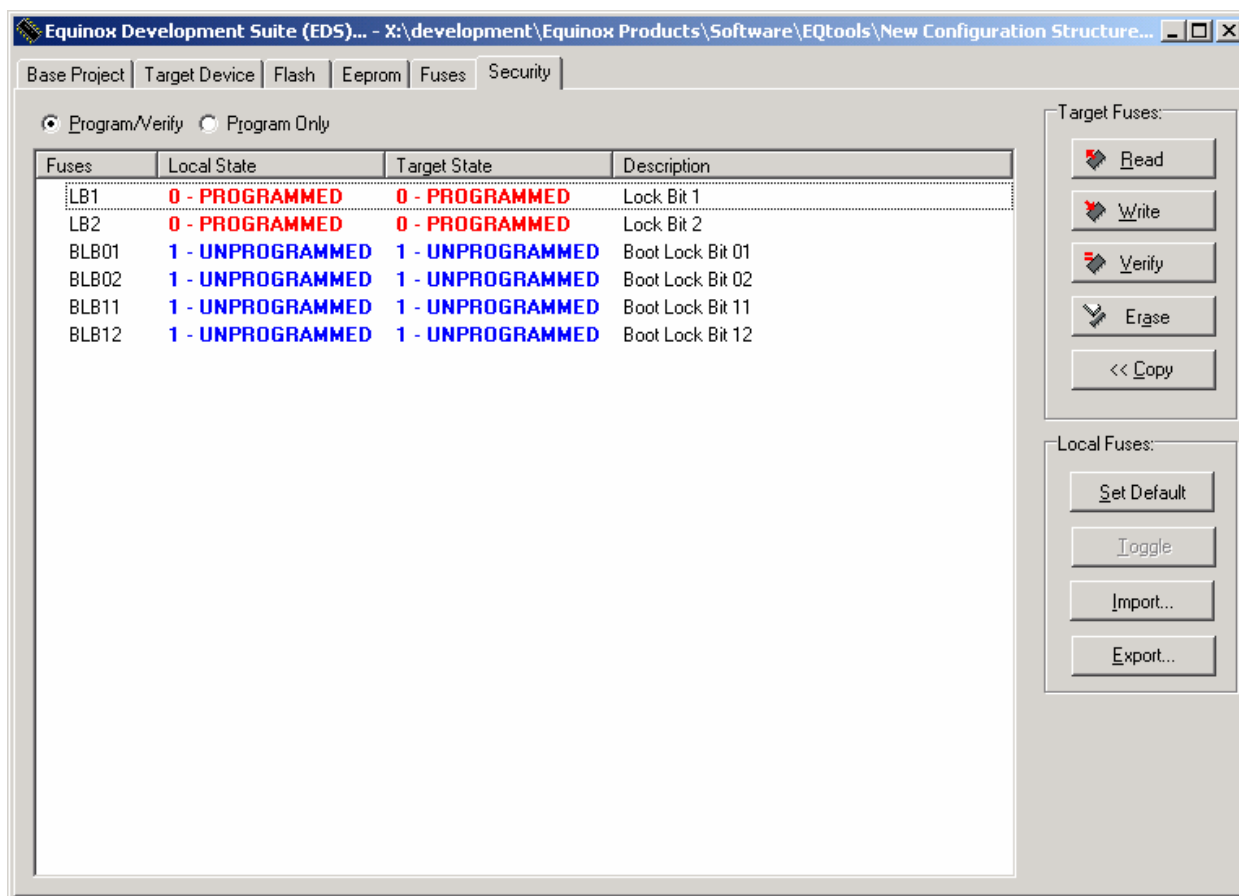
Importing the Fuse Settings from a Fuse File

To export the settings of the 'Local Fuses' column to a fuse File:

- Select the EDS <Fuses> Tab
- Click the <Import> button → a file browser appears
- Browse to and select your Fuse File (*.eff)
→ The Fuse settings are then automatically copied from the Fuse File to the 'Local Fuses' column.
- To program these settings into a Target Device, click <Write>.

5.7 <Security> tab

The EDS <Security> tab displays a list of the so-called 'Security Fuses', 'Lock Bits' or 'Security Levels' which are supported by the currently selected Target Device. Programming of these Security bits can have the effect of stopping an external programmer from reading the contents of the Target Device. The exact function of each of these fuses depends on the Target Device and so the datasheet for the device should be studied carefully before attempting to modify the Fuse Settings.





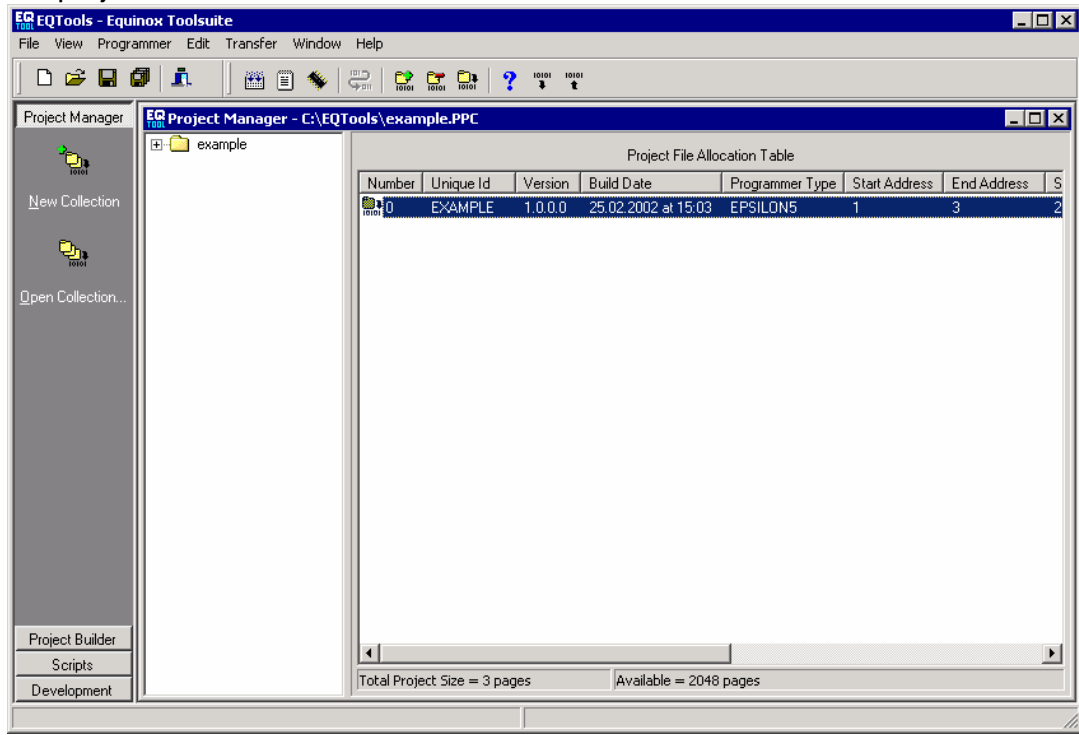

The operation of the <Security> tab is identical to that of the <Fuses> tab. Please refer to the <Fuses> tab section for a full explanation of the operation of this tab.

For Atmel AT90S and ATmega AVR microcontrollers:

- It is possible to write any of the Lock Bits from a 1 → 0 but not from a 0 → 1.
- Once you have written a Lock Bit to a '0', you must perform a Chip Erase operation in order to clear the Lock Bits back to a '1'.
- The main Lock Bits for the CODE / EEPROM area are called 'Lock Bit 1 (LB1) and 'Lock Bit 2 (LB2)'.

Any ATmega device which features a 'Boot Loader' Code Block will also have a number of 'Boot Lock' Bits which are used to protect the Boot Block in the FLASH Area of the Microcontroller. Please refer to the device datasheet for full instructions on

To add or remove projects in your project collection follow these steps:

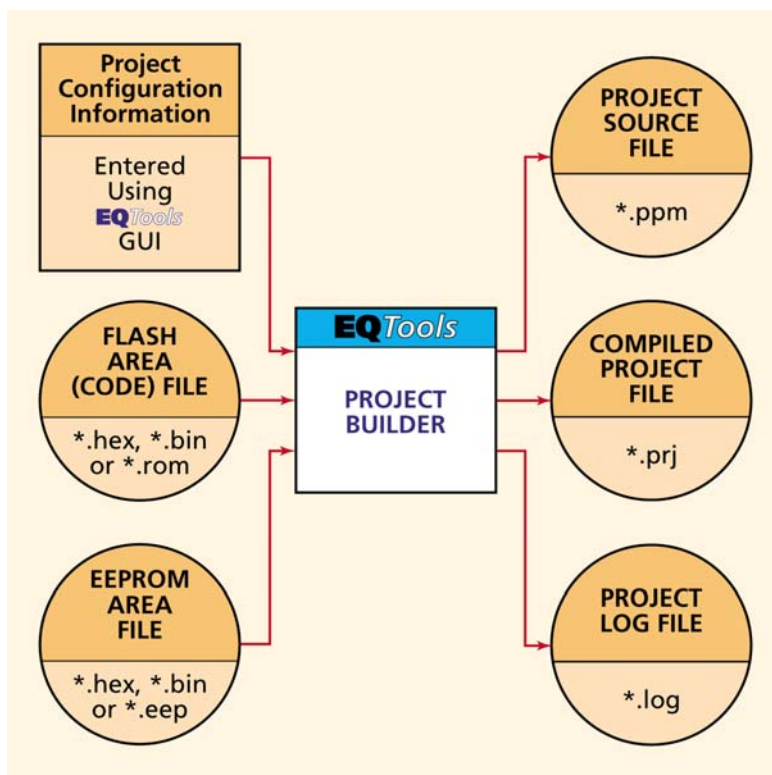
#	Action
1	Open the project collection (<Open Collection>  in the Project Manager tab)
2	Use the file selection dialog to browse to your project collection and press <Open>
3	Adding a project to your collection
3.1	Press the  <Add a project> icon
3.2	Use the file selection dialog to browse to a previously compiled project file (*.ppj) and press <Open>
3.3	The file will be added to your project collection and will be displayed as the last item in the project collection window. 
4	Removing a project from your collection
4.1	Highlight the project to be removed.
4.2	Press the  <Remove collection> icon
4.3	Confirm to remove the project.
4.4	The project collection will be rebuilt and can now be uploaded to your programmer.

6.0 Project Builder instructions

6.1 Overview

The operation of the EQTools environment is based around the creation of so-called 'Programming Projects'. Each programming project contains all the information about how to program a specified Target Device including the Target Device, power supply characteristics, RESET timings, Fuse Configurations and both the FLASH and EEPROM HEX / Binary data files. Programming Projects are created using the 'Project Builder Wizard' utility 1 and are then compiled to produce a 'Programming Project Source File (*.prj)', as detailed in fig. 6.1.

Fig. 6.1 Project Manager Overview



The steps to create and test a Programming Project are as follows:

1. Create a Programming Project Source File (*.ppm) using the 'Project Builder Wizard'
2. **Compile / build** the Project Source File (*.ppm) using Project Manager
→ a Programming Project Object file (*.prj) and Log File (*.log) are created.
3. Add the Programming Project Object file (*.prj) to a Project Collection (*.ppc)
4. Test Programming Project using the Equinox Development Suite (EDS)
5. Upload Programming Project(s) to the 'FLASH Memory Store' of a programmer and try to execute the project in 'Standalone Mode'.

These steps are automated within EQTools so that every time a Programming Project is compiled, EQTools will prompt you to add the project to a new or existing Project Collection

or to test the project in Equinox Development Suite (EDS).

File	Description
.ppm	Programmer - Project Source File <ul style="list-style-type: none">This file contains all the setup information for a Programming Project generated by EQTools.This file may be edited by the developer, but must then be re-compiled to produce a suitable project file (.prj) and then added to a Project Collection (*.ppc) to allow it to be uploaded to a programmer.ISP-PRO requires the *.ppm file for the Base Project only.
*.prj	Programmer – Project Object File <ul style="list-style-type: none">This is a single file containing all the information needed to program a target device.It is generated by compiling a *.ppm file using EQTools.This file must be added to a Project Collection (*.ppc) before being uploaded to an Equinox programmer.
*.HEX	Intel HEX File <ul style="list-style-type: none">This file contains the data which is to be programmed into either the FLASH or the EEPROM area of the target device.
*.a90	IAR AVR Compiler Output File <ul style="list-style-type: none">This file contains the data which is to be programmed into either the FLASH or the EEPROM area of the target device.The IAR compiler can be set up to output in many different formats including Intel Hex and binary.
*.bin	Binary File <ul style="list-style-type: none">This file contains the data which is to be programmed into either the FLASH or the EEPROM area of the target device.The data is stored in 'Binary' format i.e. the raw data with no address information or error checking.

6.2 Programming Project Source File (*.ppm) overview


A Programming Project Source File (*.ppm) is created using the 'Project Builder Wizard'. The Wizard will guide you through numerous tabs and will prompt you for the information below.

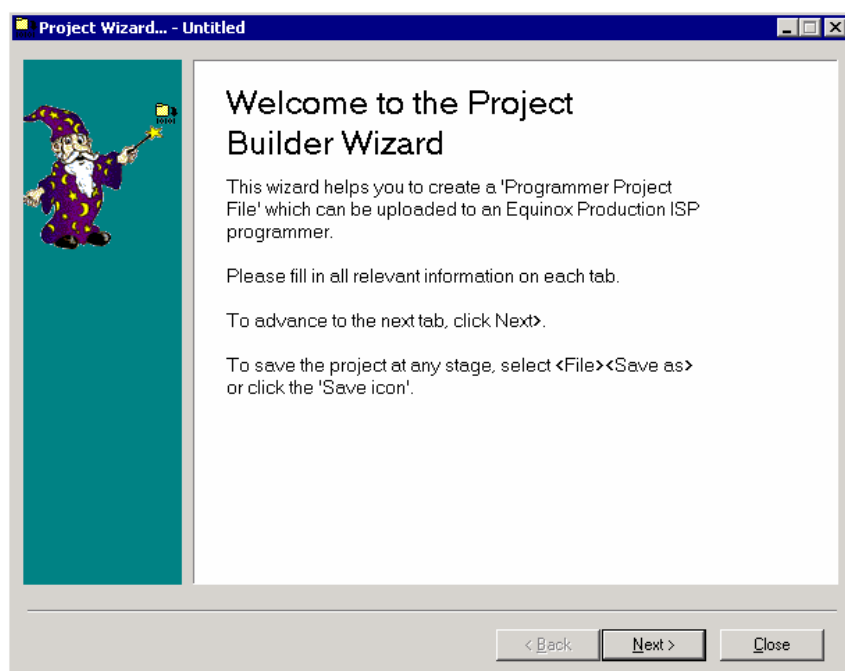
A Programming Project contains the following information:

- Selected Equinox Programmer
- Selected Programmer Control Method
- Target Connect / Disconnect Settings
- Target Device – Manufacturer's Code eg. ATmega128-16xx
- Target Device Signature(s)
- Target Voltage Settings
- Target Current Settings (if programmer powering the Target System)

- Target RESET Timing State Machine
- SPI / JTAG Frequency (for Atmel AVR and AT89S devices)
- Target Communications BAUD rate (for Atmel T89C51Rx2 and Philips P89C51Rx2/66x devices)
- FLASH Area Data File
- EEPROM Area Data File
- Target Device Configuration Fuses
- Target Device Security Settings

6.3 Project Builder – Launching the Wizard

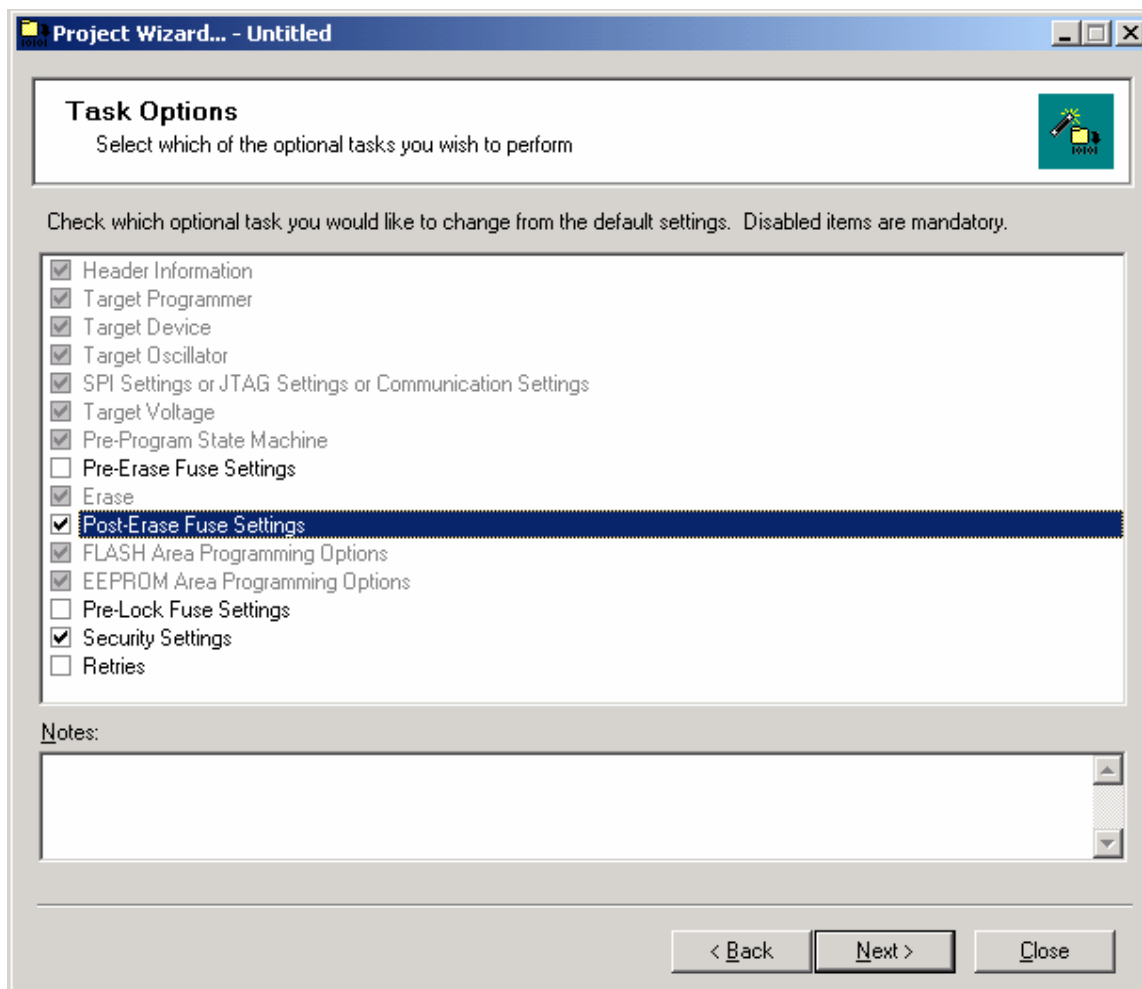
- To launch the Project Manager – Wizard:
- Launch EQTools
- Select the <Project Builder> tab from the left-hand pane
- Click the <New Project>  icon from the left-hand pane
→ The Project Wizard launches.



- Click <Next> to proceed to the next tab - <Project Tasks>.

6.4 Project Builder – Project Tasks Tab

The **<Project Tasks>** tab lists all the possible programming operations which can be performed within a Programming Project. Each 'Task' represents a separate tab in the wizard. Click **<Next>** to continue to the next tab.



Key:

- ☒ Tab is mandatory and is always displayed
- ☐ Tab is currently disabled and will not be displayed
- ☒ Tab is currently enabled and will be displayed

For Atmel AVR and AT89S projects you can usually disable the following tabs:

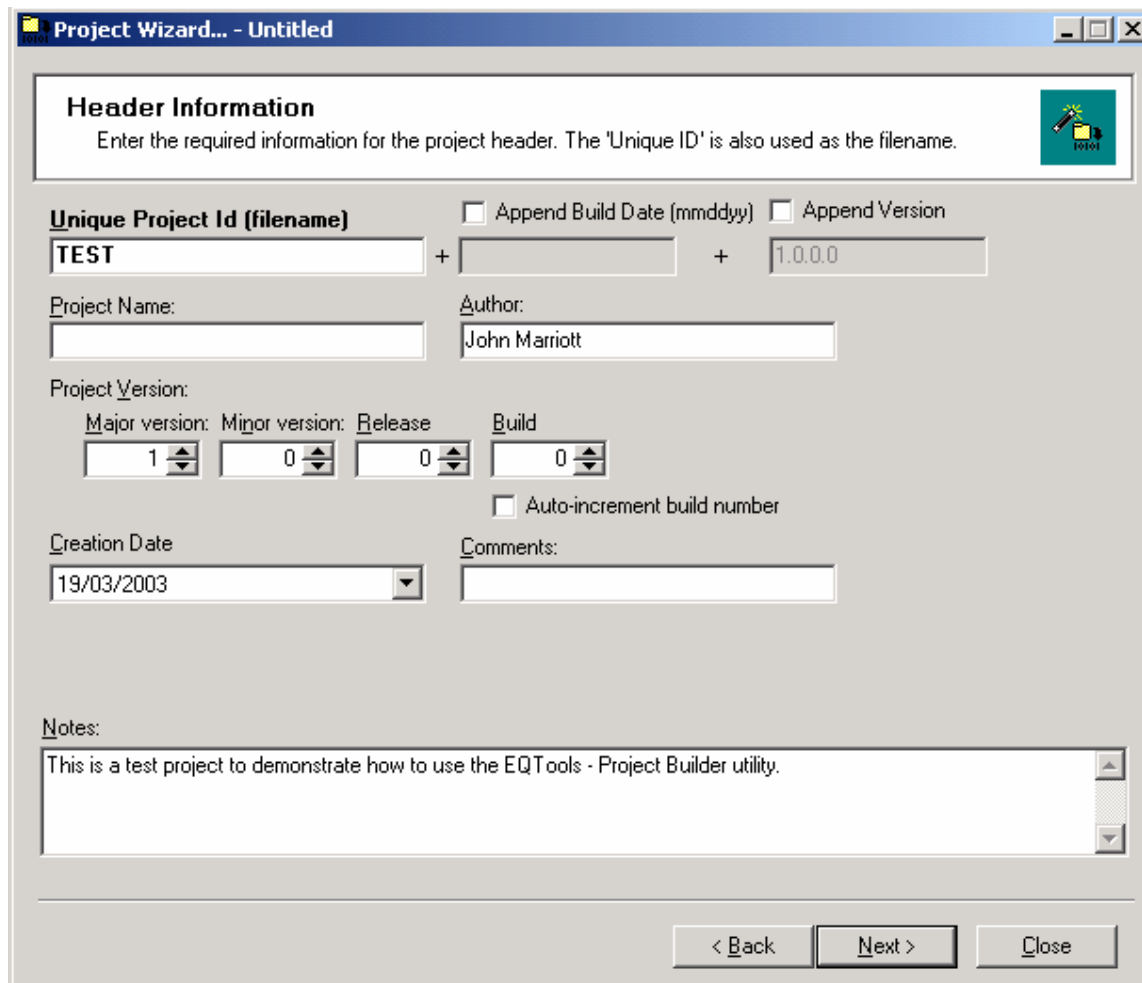
- Pre-erase Fuse Settings
- Pre-lock Fuse Settings
- Re-tries

For Atmel T89C51Rx2 and Phillips P89C51Rx2 / 66x devices, you can usually disable the following tabs:

- Pre-erase Fuse Settings
- Post-erase Fuse Settings
- Re-tries

6.5 Project Builder – Header Information Tab

This tab is used to enter a Unique Project Name' for the Project which will also be used as the project file name.



The screenshot shows the 'Project Wizard... - Untitled' window with the 'Header Information' tab selected. The window contains the following fields and controls:

- Header Information**: A title bar and a small icon in the top right corner.
- Unique Project Id (filename)**: A text field containing 'TEST'.
- Append Build Date (mmdyy)**: A checkbox that is unchecked.
- Append Version**: A checkbox that is unchecked.
- Project Name**: A text field that is empty.
- Author**: A text field containing 'John Marriott'.
- Project Version**: A section with four spinners for Major version (1), Minor version (0), Release (0), and Build (0).
- Auto-increment build number**: A checkbox that is unchecked.
- Creation Date**: A dropdown menu showing '19/03/2003'.
- Comments**: A text field that is empty.
- Notes**: A large text area containing the text 'This is a test project to demonstrate how to use the EQTools - Project Builder utility.'
- Navigation Buttons**: '< Back', 'Next >', and 'Close' buttons at the bottom right.

Instructions:

1. Enter a unique name for the project in the 'Unique Project ID' field.
 - This name can be up to 32 characters if 'Version Control' is disabled and 16 characters if 'Version Control' is enabled.
 - The name must not contain any of the following characters: '+', '-', '/', '\'.
 - This name will be used as the File Name for the project.
2. Decide whether to use Project Version Control
 - This is a method of appending the compilation data and version of each project to the Project ID.
 - To enable Version Control, tick the 'Append Build Date', 'Append Version' and 'Auto-increment build number' check boxes and set the required version number.
 - Please see section 4.9 for further details.
3. Fill in other fields (not required)

4. Click <Next> to continue to the next tab.

Key:**Unique ID / Filename**

Each 'Programmer Project' contains a unique 32-character identification string stored in ASCII format. This 'Unique ID' is used by both EQtools and the target programmer to select the correct project.

The wizard will automatically chose the 'Unique ID' as the 'Project Filename' when the project is saved. The 'Project Filename' has the file extension *.prj. The ID string can be any length from 1 to 32 characters and can contain any alphanumeric character.

Author

Enter the author of the project. e.g. Jane Doe

Project Name

This field allows you to enter a name for the project name which may be different from the 'Unique ID'. Often the 'Unique ID'

Comments

This comments field can be used for important comments which must be stored with the project when it is uploaded to the programmer.

Project Version

The version of the project / firmware may be recorded in this field.

Creation Date

This is the date on which the project was created or amended.

Header Notes

This field allows you to record comments about the project. These comments are stored in the project file, but are not stored in the actual image uploaded to the programmer.

6.6 Project Builder – Programmer Type tab

This tab allows you to select the programmer which you wish to use and the type of project.

The screenshot shows a Windows-style dialog box titled "Project Builder... - TEST". Inside, there's a section titled "Programmer and Project Type Options" with the instruction "Select the required Programmer to be used, and the type of project". Below this, there are two dropdown menus: "Programmer:" with "EPSILON5" selected, and "Project Type" with "Standalone - Keypad Control" selected. At the bottom, there's a "Notes:" label above a large text area, and three buttons: "< Back", "Next >", and "Close".

Instructions:

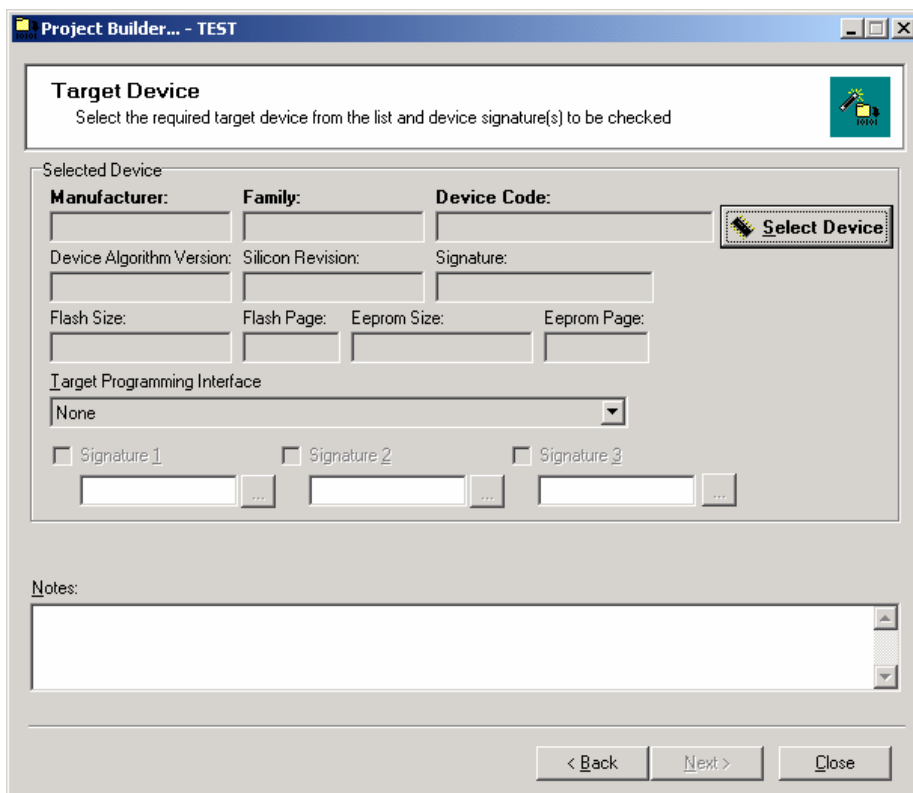
1. Select the required programmer from the drop-down list
 - The programmer type will be remembered for all subsequent projects.
2. Select the required Project Type
 - Select <Standalone – Keypad Control> if you wish to test the project in EDS or upload to a programmer to test in Standalone Mode.
 - Please refer to the table shown in fig. 6.6 for details of the other available 'Project Types'.
3. Click <Next> to continue to the next tab.

Fig. 6.6 Project Types overview

Project Type	Description
Standalone – Keypad Control	<ul style="list-style-type: none"> Project can be controlled in 'Standalone Mode' via programmer keypad. Project can also be controlled via EDS and 'ASCII Text Communications Mode'.
Standalone – Target Run	<ul style="list-style-type: none"> This is a special project type which allows the target firmware to run for a specified period of time.
Standalone – Remote I/O Control	<ul style="list-style-type: none"> Project can be controlled via the '4-Wire TTL' hardware control interface. PPM3-MK2 programmer only.
Standalone – Auto-start (Target Connect / Disconnect)	<ul style="list-style-type: none"> Project automatically senses the presence of a Target System and executes the programming project. Project automatically senses when the Target System is removed and then resets the programmer and waits for the next Target System. PPM3-MK2 programmer only.
ISP-PRO – End in Programming Mode	<ul style="list-style-type: none"> This is a special project type which should be used with projects which are called from Programming Scripts with ISP-PRO. These projects remain in Programming Mode at the end of the project so that the script does not have to re-enter Programming Mode.

6.7 Project Builder – Target Device tab

This tab allows you to select the Target Device which you wish to program.
The screen will initially have no device selected.



Project Builder... - TEST

Target Device
Select the required target device from the list and device signature(s) to be checked

Selected Device

Manufacturer: Family: Device Code: **Select Device**

Device Algorithm Version: Silicon Revision: Signature:

Flash Size: Flash Page: Eeprom Size: Eeprom Page:

Target Programming Interface
None

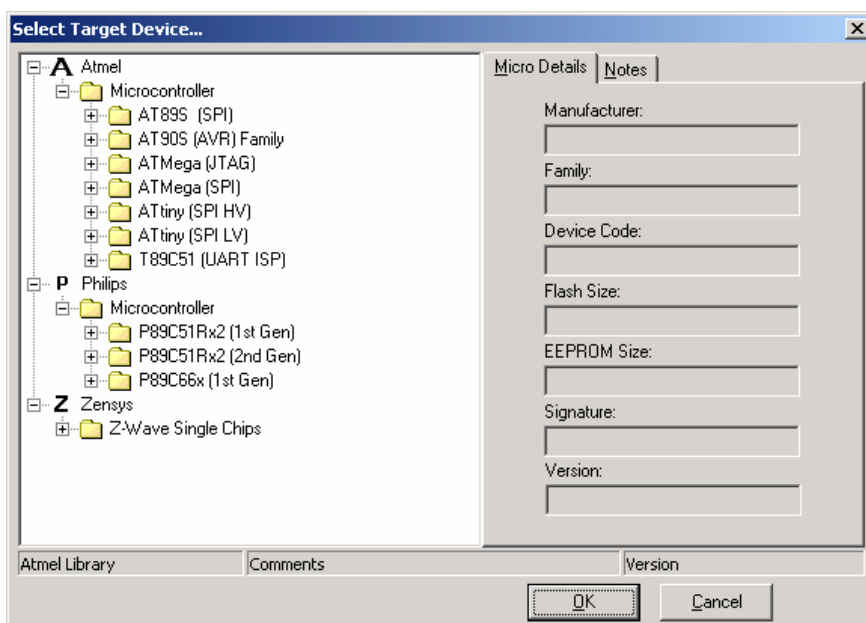
☐ Signature 1 ☐ Signature 2 ☐ Signature 3

Notes:

< Back Next > Close

Instructions:

1. Click the <Select Device> button
→ a list of the currently supported Device Manufacturer / Families is displayed.



Select Target Device...

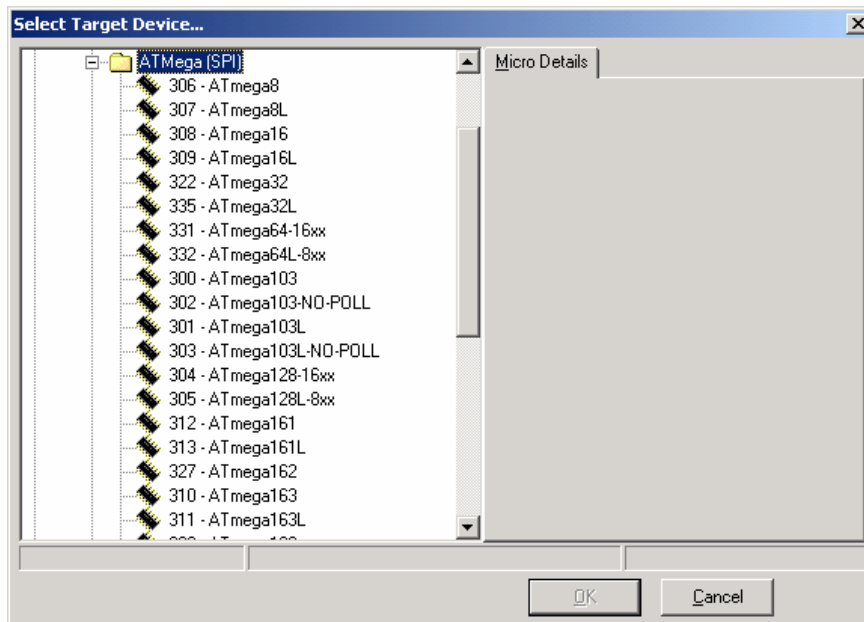
Micro Details Notes

Manufacturer: Family: Device Code: Flash Size: EEPROM Size: Signature: Version:

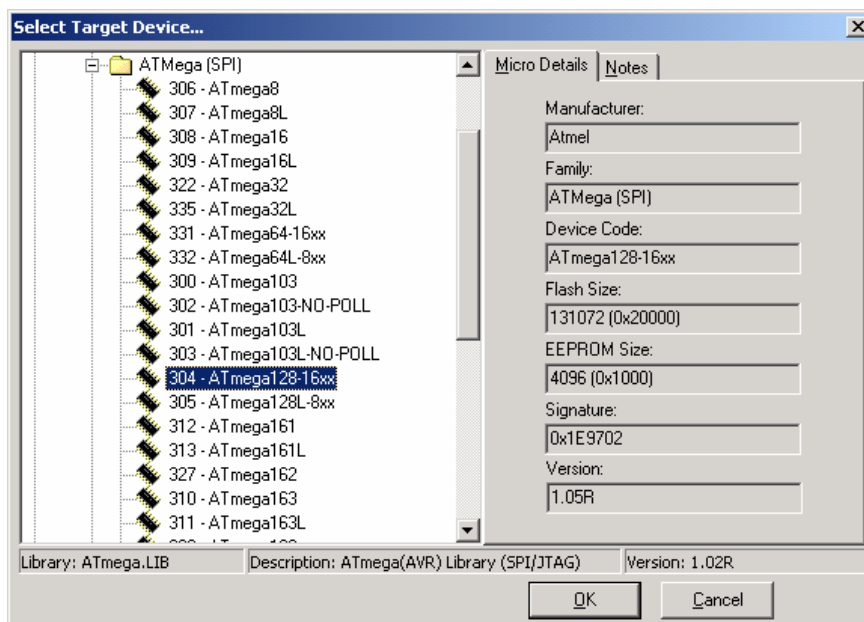
Atmel Library Comments Version

OK Cancel

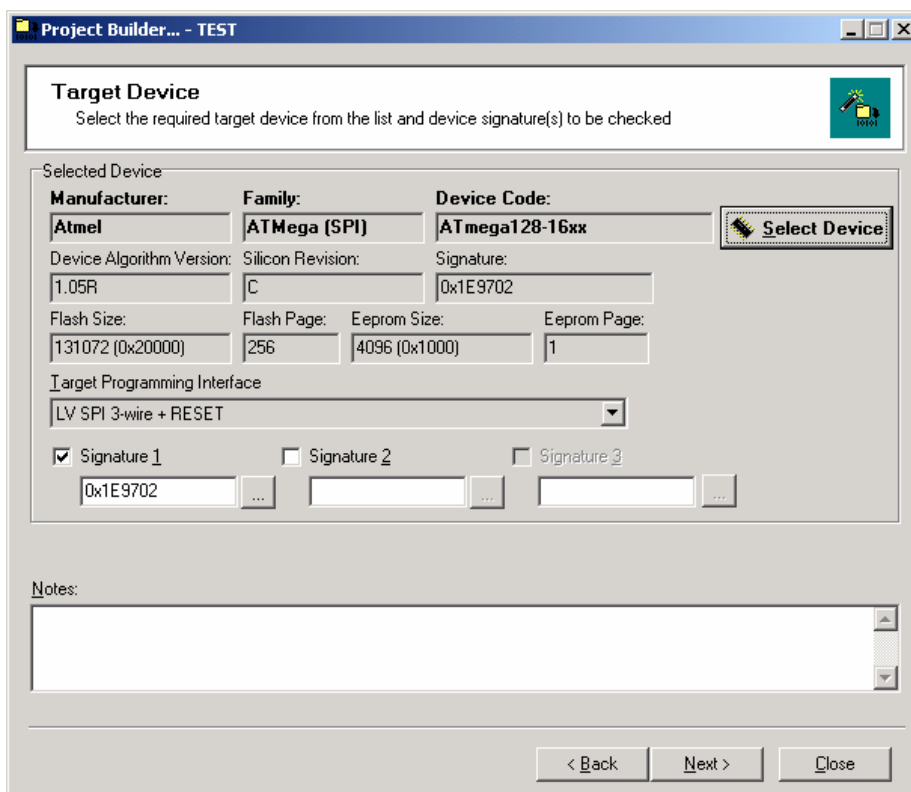
2. Select the required Device Family
 - Click the '+' box to expand the Device tree.
 - It is possible to program Atmel ATmega devices using either the SPI or JTAG algorithms. Please make sure you select the correct family – ATmega(SPI) or ATmega(JTAG).



3. Select the required Device
 - Click the required device → the details of this device are shown in the right-hand pane.
 - Click the <OK> button to confirm the selection.



4. Check the device details are correct
The device details are now displayed as shown below.



Project Builder... - TEST

Target Device
Select the required target device from the list and device signature(s) to be checked

Selected Device

Manufacturer:	Family:	Device Code:	
Atmel	ATMega (SPI)	ATmega128-16xx	
Device Algorithm Version:	Silicon Revision:	Signature:	
1.05R	C	0x1E9702	
Flash Size:	Flash Page:	Eeprom Size:	Eeprom Page:
131072 (0x20000)	256	4096 (0x1000)	1
Target Programming Interface			
LV SPI 3-wire + RESET			
<input checked="" type="checkbox"/> Signature 1	<input type="checkbox"/> Signature 2	<input type="checkbox"/> Signature 3	
0x1E9702			

Notes:

< Back Next > Close

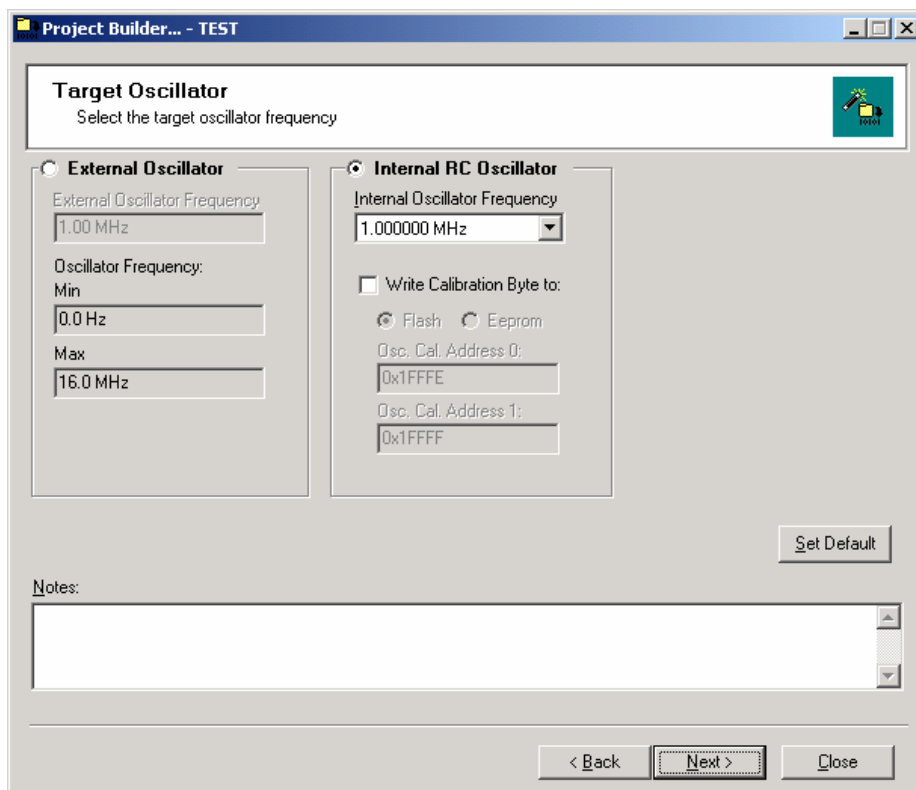
5. Add extra Device Signature(s) if required
 - Some Atmel AVR devices can have more than one signature.
 - The early AT90S devices have their quoted datasheet signature and also a so-called 'Locked Signature'.
 - If you wish to add any of these signatures so that the Programming Project will accept all possible signatures, tick the check box next to 'Signature 2' and then click the <...> button to browse for a second signature.
 - You can also manually enter a signature in the Signature 2/3 fields if required.

6.8 Project Builder – Target Oscillator tab

This tab sets up the Oscillator Frequency of the Target System. The Target Oscillator usually refers to an External Oscillator such as a crystal or ceramic resonator. However, some Atmel AVR devices also feature Internal RC Oscillators which can be configured from the screen.

Target Oscillator Frequency is used in a different way depending on the Target Device type:

- For Atmel AVR and AT89S devices, the Oscillator Frequency is required in order to calculate the maximum possible SPI frequency.
- For Atmel T89C51Rx2 devices, the Target Oscillator Frequency is not actually required. However, it is good practice to enter it here anyway.
- For Philips P89C51Rx2 and P89C66x devices, the Target Oscillator Frequency is required by the Target Device in order to calculate the correct timings to detect the first character which is used to enter programming mode.



Instructions:

1. For Atmel AVR devices which feature both Internal and External oscillators:
 - If you are planning to use the device with an External Oscillator, click the <External Oscillator> radio button and then fill in your External Oscillator Frequency in Hz.
 - If you are planning to only use one of the Internal RC Oscillators, click the <Internal RC Oscillator> radio button and then select the required Internal Oscillator from the drop-down list.
 - If you wish the programmer to read out the 'Calibration Byte' from the Target Device and then program it into TWO locations in FLASH or one location in EEPROM, tick the 'Write Calibration Byte to' check box, select either the FLASH or EEPROM radio

button and then select the address into which you wish the Calibration Byte to be written.

2. For all other devices
 - click the <External Oscillator> radio button and then fill in your External Oscillator Frequency in Hz.

Project Builder... - TEST

Target Oscillator
Select the target oscillator frequency

☒ **External Oscillator**

External Oscillator Frequency
16.00 MHz

Oscillator Frequency:
Min
0.0 Hz
Max
16.0 MHz

☐ **Internal RC Oscillator**

Internal Oscillator Frequency
1.000000 MHz

☐ Write Calibration Byte to:
☒ Flash ☐ Eeprom
 Osc. Cal. Address 0:
0x1FFFE
 Osc. Cal. Address 1:
0x1FFFF

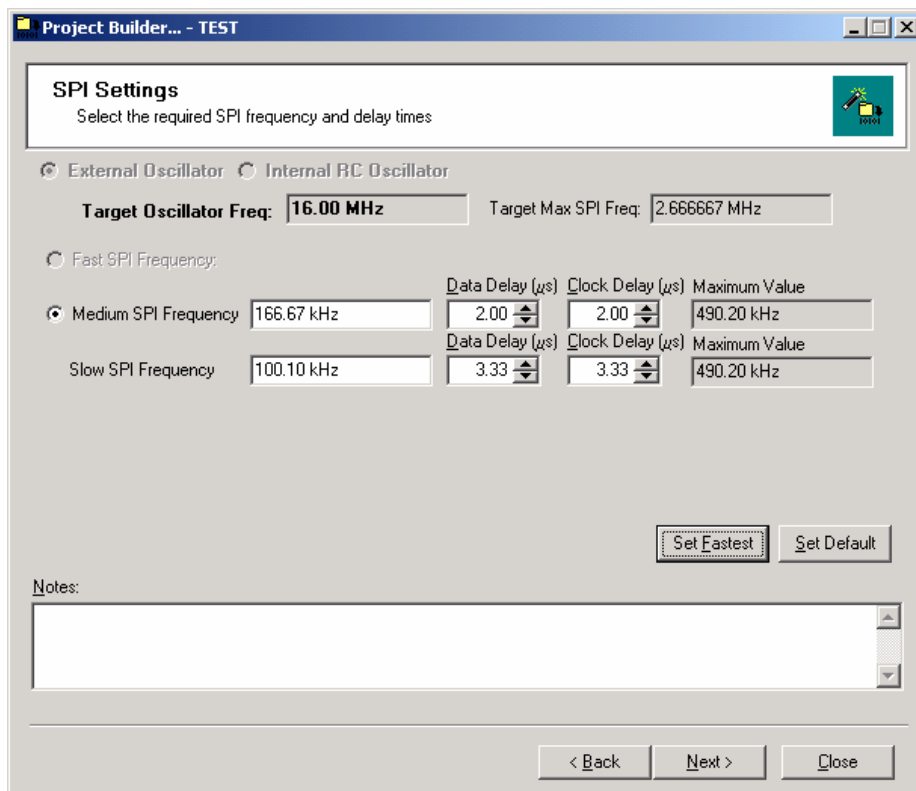
Set Default

Notes:

< Back Next > Close

6.9 Project Builder – SPI Settings tab

This tab allows you to set up the SPI Frequency which the programmer will use to communicate with the Target Device. The SPI frequency is the average frequency of the SPI clock signal on the 'PROG_SCK' pin. The frequencies quoted are estimations only as the mark / space ratio of the clock signal is not constant.



Project Builder... - TEST

SPI Settings
Select the required SPI frequency and delay times

☒ External Oscillator ☐ Internal RC Oscillator

Target Oscillator Freq: Target Max SPI Freq:

☐ Fast SPI Frequency:

	Data Delay (μs)	Clock Delay (μs)	Maximum Value
<input checked="" type="radio"/> Medium SPI Frequency	<input type="text" value="2.00"/>	<input type="text" value="2.00"/>	490.20 kHz
<input type="radio"/> Slow SPI Frequency	<input type="text" value="3.33"/>	<input type="text" value="3.33"/>	490.20 kHz

Notes:

< Back Next > Close

Instructions:

1. Adjust the 'SLOW SPI Frequency' field if required
2. Adjust the Medium / Fast SPI Frequencies by clicking the <Set Fastest> or <Set Default> buttons – see fig. 6.9.1

fig. 6.9.1 SPI Button descriptions

Button	Description
<SET FASTEST>	<ul style="list-style-type: none"> Clicking this button will calculate the fastest possible SPI frequency based on the Target Oscillator Frequency and the maximum frequency supported by the programmer. The calculated 'fastest' frequency will then be automatically entered in the Medium SPI field.
<SET DEFAULT>	<ul style="list-style-type: none"> Clicking this button will load the default SPI settings for the selected device from the Device Library. These settings should work with any device without any fine tuning. However, the programming time may be considerably longer than the <Set

	Fastest> option as a slower SPI speed will be used.
--	---

fig. 6.9.2 Definitions of Fast, Medium and Slow SPI Frequencies

SPI Frequency Setting	Description
FAST	<p>This frequency is a fixed frequency which is the fastest possible SPI frequency which the programmer can generate. This is simply too fast for some programmers so this selection is greyed out.</p> <p>This setting is used for the following SPI operations:</p> <ul style="list-style-type: none"> • Blank Check – FLASH / EEPROM • FLASH Write / Verify • EEPROM Write / Verify • CRC Check – FLASH / EEPROM
MEDIUM	<p>This frequency is a variable frequency which can be defined by the user.</p> <p>This setting is used for the following SPI operations:</p> <ul style="list-style-type: none"> • Blank Check – FLASH / EEPROM • FLASH Write / Verify • EEPROM Write / Verify • CRC Check – FLASH / EEPROM
SLOW	<p>This setting is used for the following SPI operations:</p> <ul style="list-style-type: none"> • Enter Programming Mode • Check Signature • Write Fuses / Lock Bits <p>As these operations are very fast but must be carried out correctly, it is recommended that the SLOW frequency is set to around 100 kHz for all devices. Some AVR devices clocked from an Internal 1MHz Oscillator may require an even slower setting.</p>

6.10 Project Builder – Target Power Supply Settings tab

This tab allows you to enter the characteristics of the Target Power Supply.

i. Measuring the Target Voltage

All Equinox programmers are capable of measuring the voltage applied to the 'Buffer Line Driver Circuitry' by the Target System at the start of every programming operation. This is useful to check that the Target Voltage is in fact within the voltage limits defined in the Programming Project.

To measure the Target Voltage for every programming iteration:

- Set the Voltage and Tolerance to sensible values eg. for a +5V Target Supply, set the Voltage = 5.0V and the tolerance = 500mV.
- Tick the 'Verify Buffer Voltage' check box
- The programmer will now read the Target Voltage at the beginning of every programming iteration and compare it with the Voltage / Tolerance specified in the project.
- If the voltage is within the limits specified then the check will pass and the programmer will carry on with the programming cycle.
- If the voltage check fails, the programming operation will be immediately aborted and the programmer will display the error message: Buffer Supply is out of limits

ii. Supplying a controlled voltage to the Target System

The PPM3 MK2 programmer is capable of supplying a controlled voltage in the range of 3.0 – 5.0V from the programmer to the Target System. This allows the programmer to sequence power to Target System at a specified voltage and maximum current level under Programming Project control. This scenario is better from an ESD point of view because the programmer 'I/O Driver Circuitry' will always be powered up at the same time as the Target System.

To output a controlled voltage to the Target System:

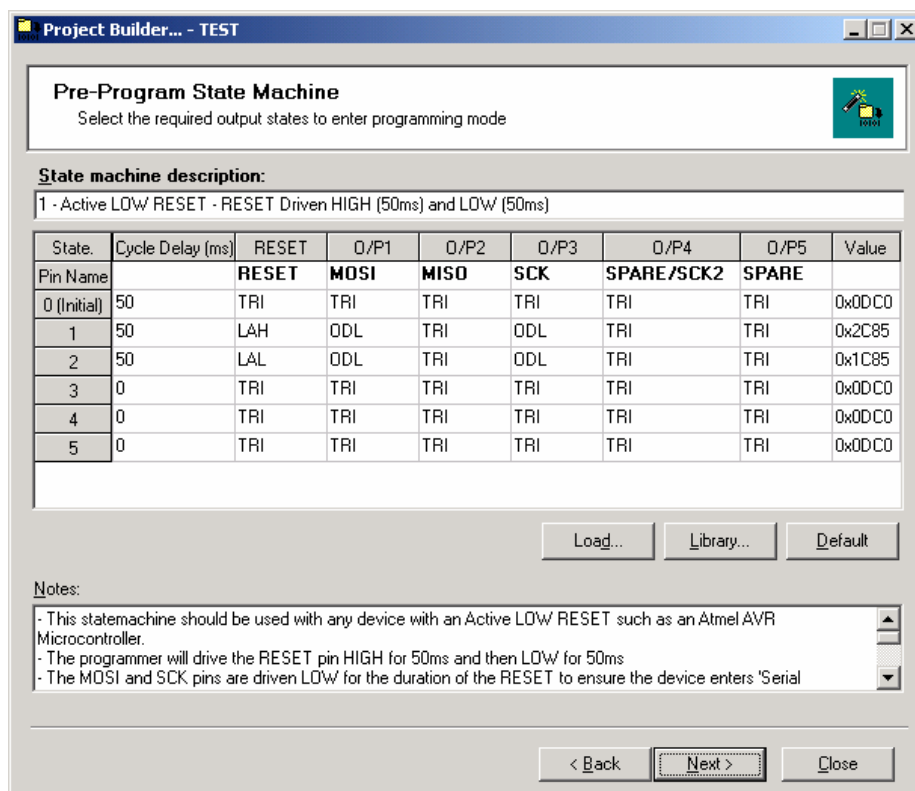
- Set the Voltage and Tolerance to sensible values eg. for a +5V Target Supply, set the Voltage = 5.0V and the tolerance = 500mV.
- In the 'Programmer Controlled Power Supply' section, tick the 'Target/Interface Module powered by PPM' check box.
- Enter the worst case inrush current value in the 'Maximum Current' field.
- Enter a suitable 'Current Stabilise' Time.
- Enter a suitable 'Voltage Settling' Time.

Fig. 6.10 Explanation of power supply terminology

Parameter	Units	Description
Voltage	V	<ul style="list-style-type: none"> • This is the nominal DC value of the target system power rail which supplies the target microcontroller. • The programmer measures this voltage using an A/D input connected to the 'Target Vcc' line. • It is therefore imperative that even if the programmer is not to be powered from the target system, the 'PROG_VCC' pin must be connected to the Vcc of the target system.
Tolerance	mV	<ul style="list-style-type: none"> • This is the allowed voltage tolerance when measuring the 'Target Power Supply Voltage'. • If the programmer is not supplying the 'Target Power Supply Voltage' and you do not wish to check the target Vcc, set the tolerance value to a high value eg. 1000 mV.
Stabilise time	ms	<ul style="list-style-type: none"> • This is the time allowed by the programmer for the 'Target Power Supply Voltage' to reach its nominal value. • If the programmer is supplying the target Vcc, then the stabilise time should be set to eg. 50 ms. If the target system power supply is to supply the 'Target Vcc', then the 'Stabilise Time' should be chosen to suit your target system power supply characteristics.

6.11 Project Builder – Pre-programming State Machine tab

This tab allows you to set up the sequencing of the programmer I/O pins which is used to force the Target Device into Serial Programming Mode. A default setting is automatically selected when the Target Device is first selected. This can then be manually modified or a completely different state machine can be loaded from the library.



Project Builder... - TEST

Pre-Program State Machine
Select the required output states to enter programming mode

State machine description:
1 - Active LOW RESET - RESET Driven HIGH (50ms) and LOW (50ms)

State.	Cycle Delay (ms)	RESET	O/P1	O/P2	O/P3	O/P4	O/P5	Value
Pin Name		RESET	MOSI	MISO	SCK	SPARE/SCK2	SPARE	
0 (Initial)	50	TRI	TRI	TRI	TRI	TRI	TRI	0x0DC0
1	50	LAH	ODL	TRI	ODL	TRI	TRI	0x2C85
2	50	LAL	ODL	TRI	ODL	TRI	TRI	0x1C85
3	0	TRI	TRI	TRI	TRI	TRI	TRI	0x0DC0
4	0	TRI	TRI	TRI	TRI	TRI	TRI	0x0DC0
5	0	TRI	TRI	TRI	TRI	TRI	TRI	0x0DC0

Load... Library... Default

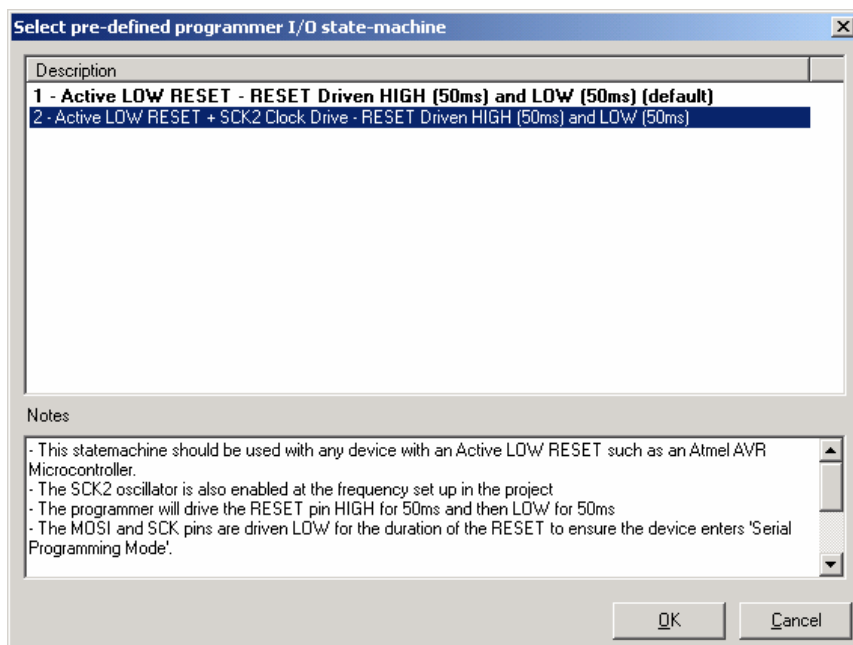
Notes:

- This statemachine should be used with any device with an Active LOW RESET such as an Atmel AVR Microcontroller.
- The programmer will drive the RESET pin HIGH for 50ms and then LOW for 50ms
- The MOSI and SCK pins are driven LOW for the duration of the RESET to ensure the device enters 'Serial

< Back Next > Close

To load a new State Machine from the Library:

- Select the <Pre-program State Machine> tab
- Click the <Library> button
→ a list of pre-defined I/O State Machines is displayed.
- Select a State Machine and click <OK>
→ the settings for this State Machine are now transferred to the main <Pre-Programming State Machine> tab.



6.12 Project Builder – Fuse Settings tabs

Overview

Many programmable devices feature 'Configuration Fuses' which are used to control on-chip functions of the device such as the Oscillator selection, Start-up / Reset Timing, on-chip Brownout Detector control etc. These fuses can usually only be programmed by an external programmer in 'Parallel' or 'Serial' programming mode.

It is possible to program the Configuration Fuses of a Target Device at any of three separate stages during a Programming Project:

- Before the Chip Erase Operation (Pre-Erase Fuse tab)
- Immediately after the Chip Erase Operation (Post-Erase Fuse tab)
- Immediately before writing the Security Fuses (Pre-Lock Fuse tab)

In most applications it is only required to program the Configuration Fuses once during the programming cycle. As a general rule of thumb, the <Post-Erase Fuse> tab is used for all Atmel AVR and AT89S device. The <Pre-Lock Fuse> tab is used for all Atmel T89C51Rx2 and Philips P89C51Rx2 / 66x devices.

Device Family	Pre-Erase Fuse tab	Post-Erase Fuse tab	Pre-Lock Fuse Tab	Comments
AT90S		YES		<ul style="list-style-type: none"> • Always use the <Post-Erase Fuse> tab
ATmega	OPT	YES	OPT	<ul style="list-style-type: none"> • See ATmega note
ATtiny		YES		<ul style="list-style-type: none"> • Always use the <Post-Erase Fuse> tab
AT89S		YES		<ul style="list-style-type: none"> • Devices do not have any Fuses.
T89C51Rx2			YES	<ul style="list-style-type: none"> • Always use <Pre-Lock Fuse> tab
T89C51CCxx			YES	<ul style="list-style-type: none"> • Always use <Pre-Lock Fuse> tab
P89C51Rx2			YES	<ul style="list-style-type: none"> • Always use <Pre-Lock Fuse> tab
P8966x			YES	<ul style="list-style-type: none"> • Always use <Pre-Lock Fuse> tab

Key:

YES – This Fuse tab is enabled by default.

OPT – This Fuse tab can be used for an optional special function.

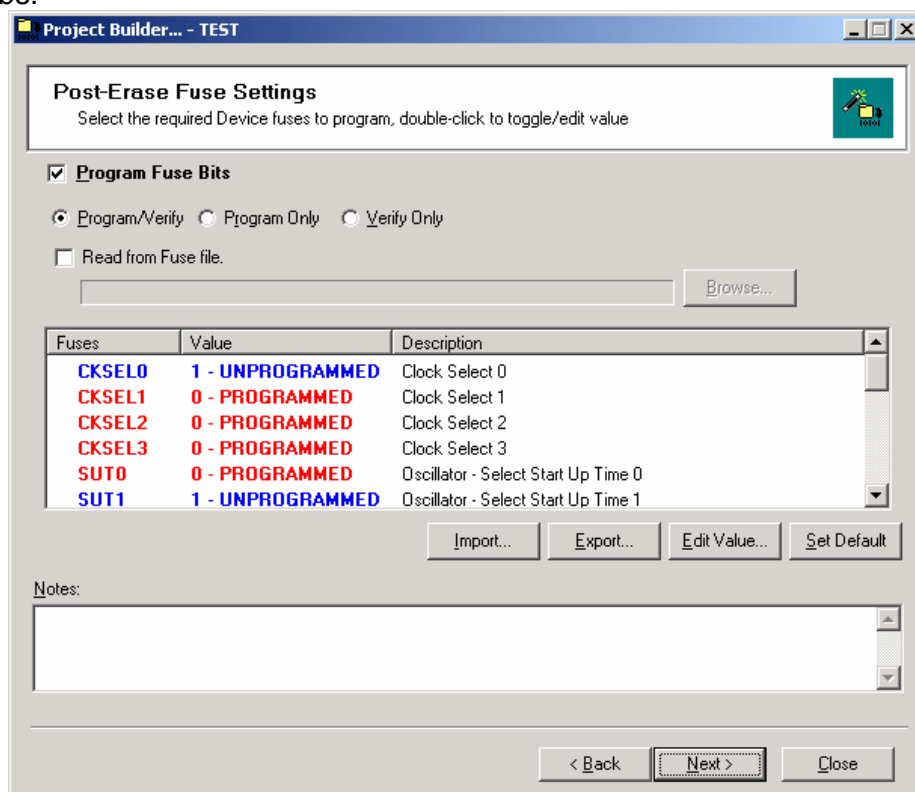
Atmel ATmega Fuses

The Atmel ATmega devices feature 'Clock Selection Fuses' which determine which oscillator the device will run from.

- If the device oscillator is simply to be swapped from Internal to External Oscillator then use the <Post Erase Fuse> tab only.
- If the Target Device is to be run from an External Oscillator source during programming only, use the <Post-Erase Fuse> tab to select External oscillator and then use the <Pre-Erase Fuse> tab to set the oscillator back to Internal Oscillator.
- If the device features an EESAVE fuse and it is enabled, then the <Pre-Erase Fuse> tab can be used to disable the fuse before a Chip Erase operation is performed. The EESAVE fuse must then be re-enabled in either the <Post-Erase Fuse> or <Pre-Lock Fuse> tabs.

Fuse tab description

The Fuse tab is shown below. This tab is identical for Pre / Post Erase and Pre-Lock Fuse Tabs.



To edit the value of a Boolean Fuse:

- Select the required Fuse by clicking once on it.
- Double-click the Fuse to alternate the value between '1' and '0'.

or

- Select the Fuse and then click the <Edit Value> button

To edit a HEX Fuse:

- Select the required Fuse by clicking once on it.
- Double-click the Fuse or click the <Edit Value> button
- Enter the required HEX value and then click <OK>

The function of the buttons on this tab are detailed in the table below.

#	Button	Description / Purpose
1	Set Default	<ul style="list-style-type: none"> This button loads the default values for each Fuse as defined in the Target Device manufacturer's datasheet. These settings usually represent the state of the fuses of a virgin device (i.e. not programmed before).
2	Edit Value	<ul style="list-style-type: none"> This button allows you to edit the value of a Fuse. It is also possible to edit a Fuse by simply double-clicking on the Fuse in the list.
3	Import	<ul style="list-style-type: none"> This button allows a pre-specified list of Fuse Settings to be loaded from an Equinox Fuse File (*.eff) into the Fuse Settings of this tab. The Fuse File must have already been created by exporting a Fuse File either in this window or in the EDS <Fuses> tab.
4	Export	<ul style="list-style-type: none"> This button exports the Fuse settings as declared on this tab to an Equinox Fuse File (*.eff). This Fuse File can then be imported back into any Programming Project in the Project Manager – <Fuse Settings> tab or into the EDS <Fuses> tab. This is useful to copy the known fuse settings from a device to a file.

! Warning!

It is possible to lose contact with the target device by setting certain combinations of `Fuse Settings.

For Atmel ATmega devices:

Please carefully check settings for the `CLKSEL' and `RSTDISBL' fuses BEFORE programming them into a target device. Once these fuses are incorrectly set, it may be necessary to manually apply an oscillator to the target device in order to re-program it again.

For Philips P89C51Rx2 and 66x devices:

Make sure that the 'BVA' fuse is not unintentionally written to a value other than 0xFC as the device will then no longer communicate with an ISP programmer.

Exporting / Importing Fuse Settings to / from File

One of the new power features of EQTools is the ability to export the Fuse Settings for a Programming Project to a Fuse File (*.eff). This allows the settings for all the fuses to be contained in one Fuse File which can then be imported into any of the Fuse tabs in Project Manager or in the <Fuses> tab in EDS.

Exporting the Fuse Settings to a Fuse File

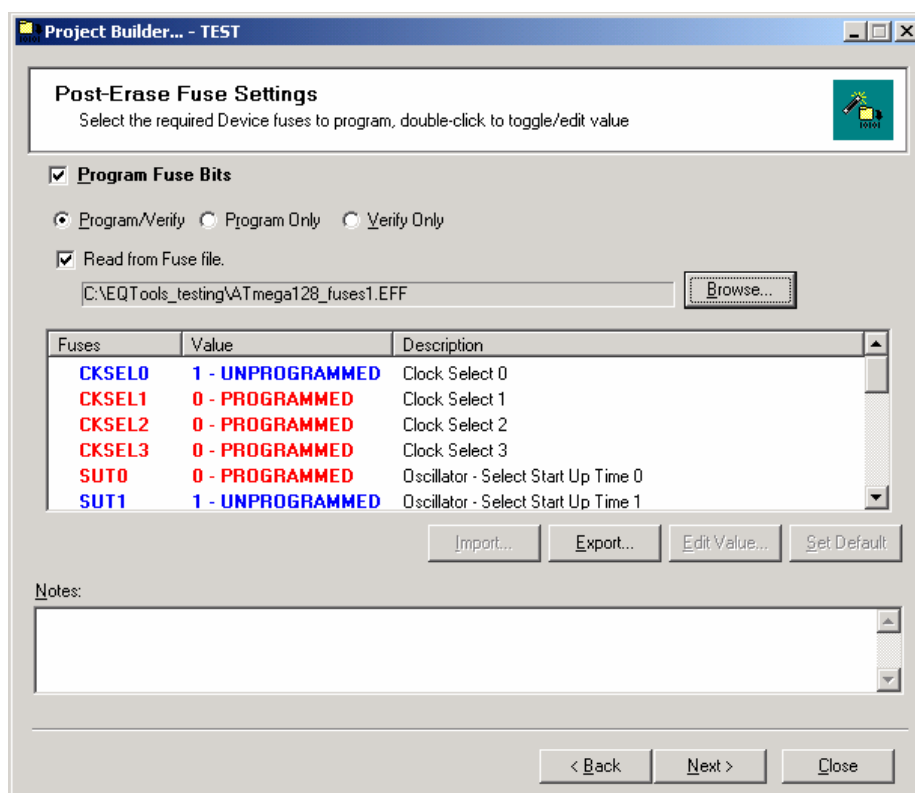
To export the settings to a fuse File:

- Select the required <Fuse> tab
- Set up the Fuses to the correct values for your Target Device.
- Click the <Export> button → a file browser is displayed.
- Enter a suitable name for your Fuse File eg. **project_fuses.eff**
- Click <Save> → The 'Local Fuses' column is transferred to your specified Fuse File.

Importing the Fuse Settings from a Fuse File

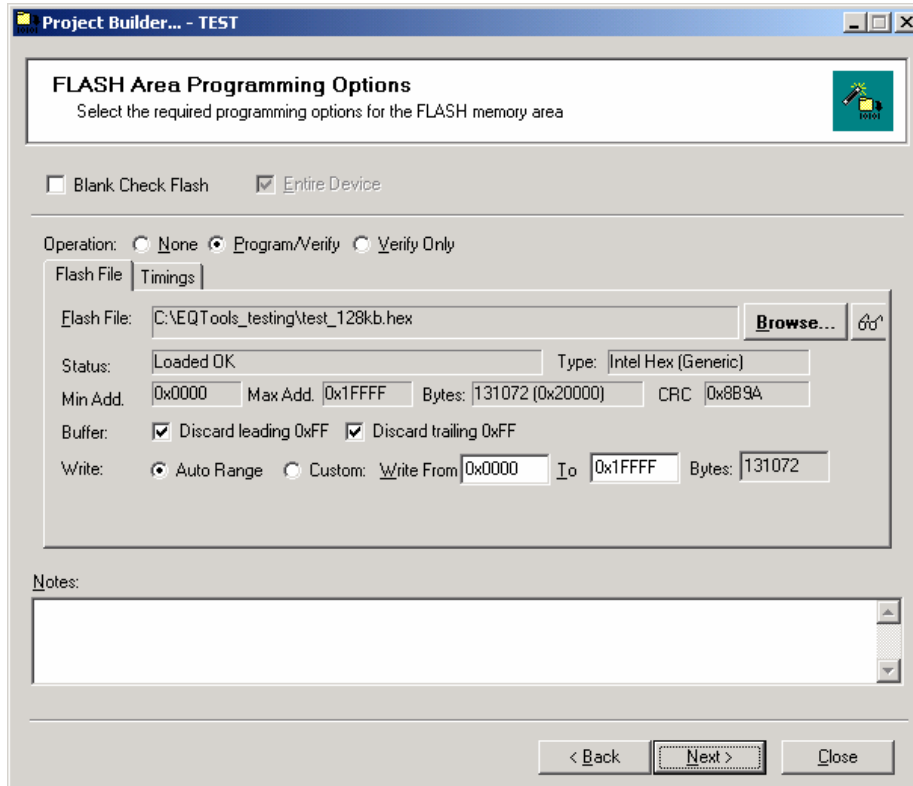
To export the settings of the 'Local Fuses' column to a fuse File:

- Select the required <Fuse> Tab
- Click the <Import> button → a file browser appears
- Browse to and select your Fuse File (*.eff)
→ The Fuse settings are then automatically copied from the Fuse File to the <Fuse> tab
- These Fuse settings will be used when the project is compiled.



6.13 Project Builder – FLASH / EEPROM tab

This tab is where the HEX or Binary File which is to be programmed into the FLASH or EEPROM area of the Target Device is set up. The FLASH / EEPROM 'Page Write' timings and 'Read-back Polling' options are also configured using this tab.

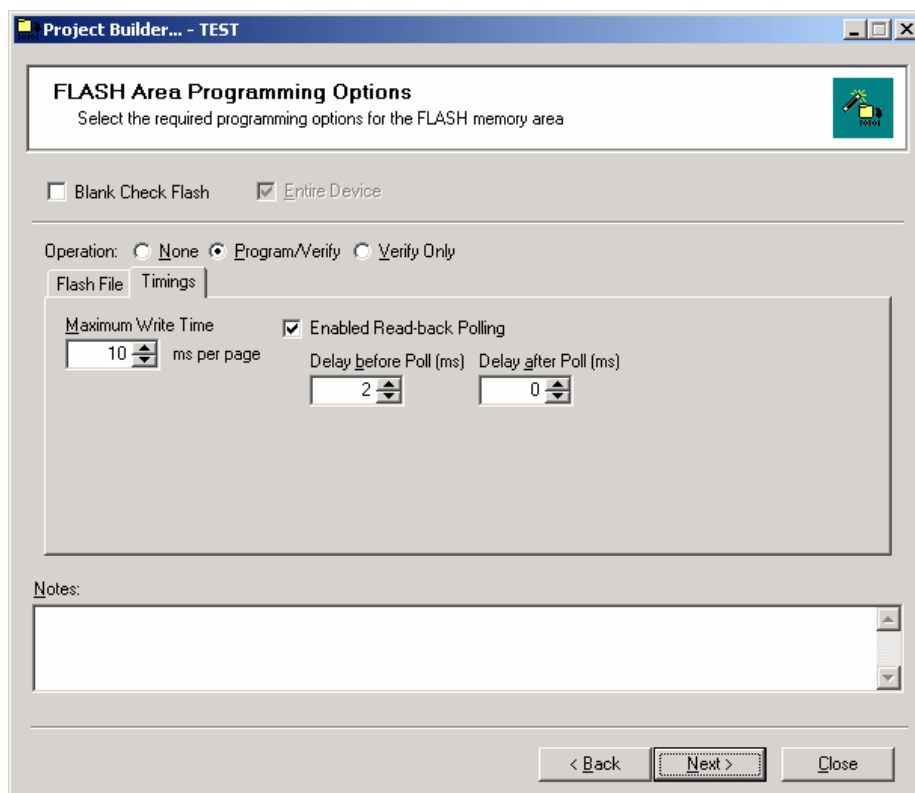


To configure a file to be programmed into the FLASH / EEPROM area:

- Click the <Program/Verify> radio button
- Click the <Browse> button → a file browser is displayed.
- Browse to and select the required HEX or Binary file. → the file is loaded and the file parameters are displayed.
- Select whether any leading or trailing '0xFF' locations should be removed to speed up programming.
- Select the Target Device address range which the file should be written to.
- When all parameters are correct, click <Next> to continue to the next tab.

Timings tab

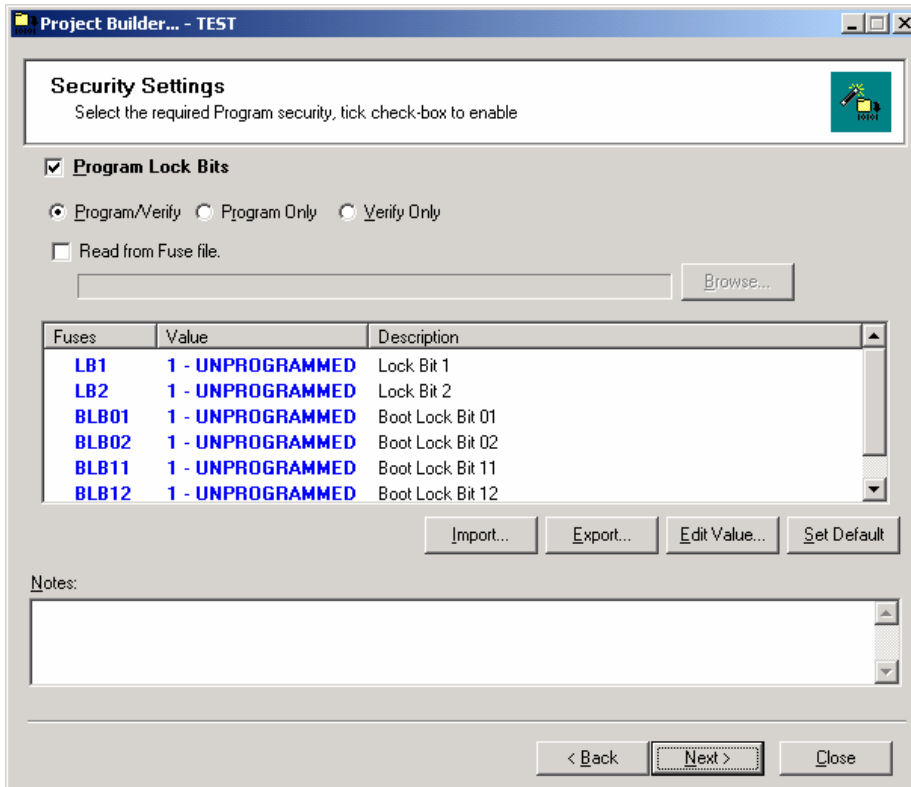
The timings used for programming the Target Device can be configured on this tab. When the Target Device and Target Voltage are specified in the project, the timings in this tab are automatically populated to the default values for the selected device and voltage level. If any programming issues arise, it may be necessary to alter these timings. Please contact Equinox for help with any further configuration of the timings.



6.14 Project Builder – Security Settings tab

This tab is used to configure the Security Fuses or Security Level of the Target Device. These Fuses allow the memory areas of the Target Device to be protected from either further verification or reading out of the entire memory area.

The operation of this tab is identical to the <Fuse> tabs. Please refer to the <Fuse> tab section for a description of how to use this tab.



Project Builder... - TEST

Security Settings
Select the required Program security, tick check-box to enable

☒ **Program Lock Bits**

☒ Program/Verify ☐ Program Only ☐ Verify Only

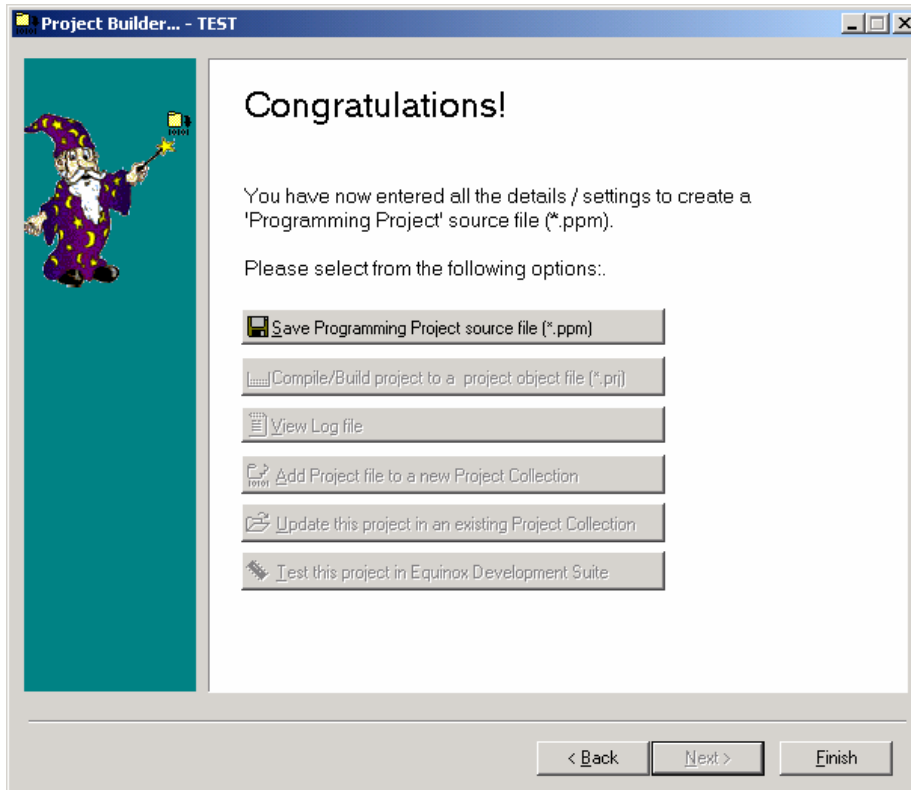
☐ Read from Fuse file.

Fuses	Value	Description
LB1	1 - UNPROGRAMMED	Lock Bit 1
LB2	1 - UNPROGRAMMED	Lock Bit 2
BLB01	1 - UNPROGRAMMED	Boot Lock Bit 01
BLB02	1 - UNPROGRAMMED	Boot Lock Bit 02
BLB11	1 - UNPROGRAMMED	Boot Lock Bit 11
BLB12	1 - UNPROGRAMMED	Boot Lock Bit 12

Notes:

6.15 Project Builder – Saving and compilation



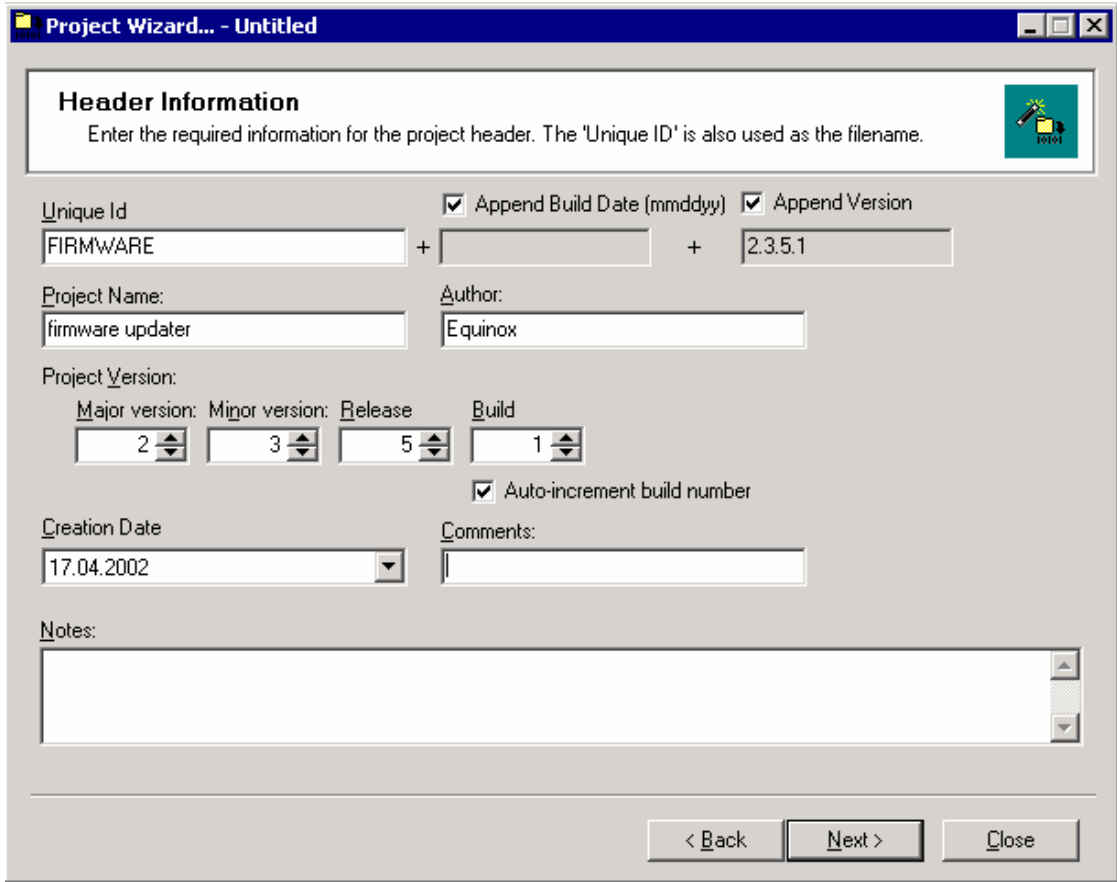

At the end of the Project Builder Wizard, you will be presented with the screen below which lists the available options.



1. Click the <Save Programming Project source File> button
 - The Programming Project is saved with the file name declared in the 'Unique ID' field on the <Header> tab.
 - The rest of the options are now selectable.

6.16 Using version control

It is often necessary to store information about the firmware version in your projects. You can store this information in the Header Information of your project. Just follow the instructions detailed below.

#	Action
1	Open a project () or create a new project ()
2	Advance to the Header Information tab
3	Enter the version information of the firmware in <Major version>, <Minor version> and <Release>. Leave <Build> blank. 
4	Enable the features <Auto-increment build number>, <Append Build Date (mmddyy)> and <Append Version>
5	Save and compile the project () or in case of a new project fill in the required tabs and press <Save> and <Compile>

The build number will be incremented every time you compile this project and can therefore be used to trace any changes made in the project.

The project itself will be called <UniqueID>_<BuildDate>_<CompleteVersion> (e.g. FIRMWARE_170402_2.3.5.1). This name is displayed in the project collection window, on the LCD of an FS2000 or PPM3, in the LabView Control and in the Interface Database.

7.0 Project / Script File explanations

7.1 Overview

EQTools generates a whole host of different files during the development of a Programming Project, Project Collection and a Programming Script. Many of these files are 'source' files e.g. *.ppm, *.esw, *.hex and are only required by the developer to maintain/edit the project. These files are NOT required by the production software ISP-PRO and should therefore NOT be included in any files sent to a 'production facility' unless you wish the 'production operators' to be able to edit/re-compile the project/script files.

7.2 EQTools – Source files & output files overview

The table below lists all the possible files which EQTools either requires as input files or generates as output files. These files should all be stored in your working project directory. The only exception is the Interface Database, which can be found in the **program files\Equinox\ispproldb** directory by default.

File	Description
*.esw	Equinox Script File (source file) <ul style="list-style-type: none"> This is the source file which is used by the EQTools – Script Builder utility. When the *.esw file is compiled it produces a *.esf file which is what the programmer actually executes.
.esf	Equinox Script File (object file) <ul style="list-style-type: none"> This is the actual programming script file which is executed by each programming channel within ISP-PRO. The file is generated by compiling an Equinox Script Wizard File (.esw) using EQTools.
*.fus	Fuse / Lock Bit Description File <ul style="list-style-type: none"> This file is produced by the Script Wizard when a <Fuse Write> or <Fuse Verify> task is enabled. These files contain a map of the settings of all fuse/lock bits as defined by the <Fuse Write> or <Fuse Verify> task and are therefore required for running a script within ISP-PRO.
.ppm	Programmer - Project Source File <ul style="list-style-type: none"> This file contains all the setup information for a Programming Project generated by EQTools. This file may be edited by the developer, but must then be re-compiled to produce a suitable project file (.prj) and then added to a Project Collection (*.ppc) to allow it to be uploaded to a programmer. ISP-PRO requires the *.ppm file for the Base Project only.
*.prj	Programmer – Project Object File <ul style="list-style-type: none"> This is a single file containing all the information needed to program a target device.

	<ul style="list-style-type: none"> It is generated by compiling a *.ppm file using EQTools. This file must be added to a Project Collection (*.ppc) before being uploaded to an Equinox programmer.
.ppc	Programmer – Project Collection <ul style="list-style-type: none"> This is a single file containing a collection of up to 64 independent programming projects (.prj). This file must be uploaded to the programmer FLASH Memory Store before running the programming script.
*.mdb	Interface Database This is the Interface Database used by EQTools and ISP-PRO when running programming scripts. The default location for this database is program files\isppro\db directory.
*.HEX	Intel HEX File <ul style="list-style-type: none"> This file contains the data which is to be programmed into either the FLASH or the EEPROM area of the target device.
*.a90	IAR AVR Compiler Output File <ul style="list-style-type: none"> This file contains the data which is to be programmed into either the FLASH or the EEPROM area of the target device. The IAR compiler can be set up to output in many different formats including Intel Hex and binary.
*.bin	Binary File <ul style="list-style-type: none"> This file contains the data which is to be programmed into either the FLASH or the EEPROM area of the target device. The data is stored in 'Binary' format i.e. the raw data with no address information or error checking.
*.map	Map File <ul style="list-style-type: none"> This file contains the declarations for all FLASH or EEPROM constants used within the Script Wizard. These declarations are used in the 'Script Tasks' dialog for read/writing unique information from/to the target device.

7.3 Files required by ISP-PRO to execute a programming script

The table below details the actual files which are required by ISP-PRO in order to execute a programming script. These files, with the exception of the Interface Database, should all be copied into the specified **lscripts** directory.

File	Description
*.esf	Equinox Script File <ul style="list-style-type: none"> If you are using more than one script file, then you need to copy all these *.esf files to the same lscripts directory.
*.prj	Programmer – Project Object File <ul style="list-style-type: none"> Only the 'Base Project' *.prj file is required. If you have other projects contained in the Project collection (*.ppc), you do NOT need the relevant *.prj files to execute the script.
.ppc	Programmer – Project Collection <ul style="list-style-type: none"> This is a single file containing a collection of up to 64 independent programming projects (.prj). This file must be uploaded to the programmer FLASH Memory Store before running the programming script.
*.fus	Fuse / Lock Bit Description File <ul style="list-style-type: none"> This file is produced by the Script Wizard when a <Fuse Write> or <Fuse Verify> task is enabled. These files contain a map of the settings of all fuse/lock bits as defined by the <Fuse Write> or <Fuse Verify> task and are therefore required for running a script within ISP-PRO.
.mdb	Interface Database <ul style="list-style-type: none"> This is the Interface Database used by EQTools and ISP-PRO when running programming scripts. The default location for this database is program files\ispproldb directory and the default alias to this database is ac_isppro. You should not need to change this file unless you have added custom fields to the database. In this case the new custom database would have to be installed on any PC which is to run the script. You should not need to send the Interface Database (.mdb) file to the Production Facility as they should have exactly the same database if they have installed the same version of ISP-PRO. It is possible for a Script to validate that the correct database is being used. This feature can be enabled in the <Validate Database> section of the EQTools – Script Wizard <Database> tab.

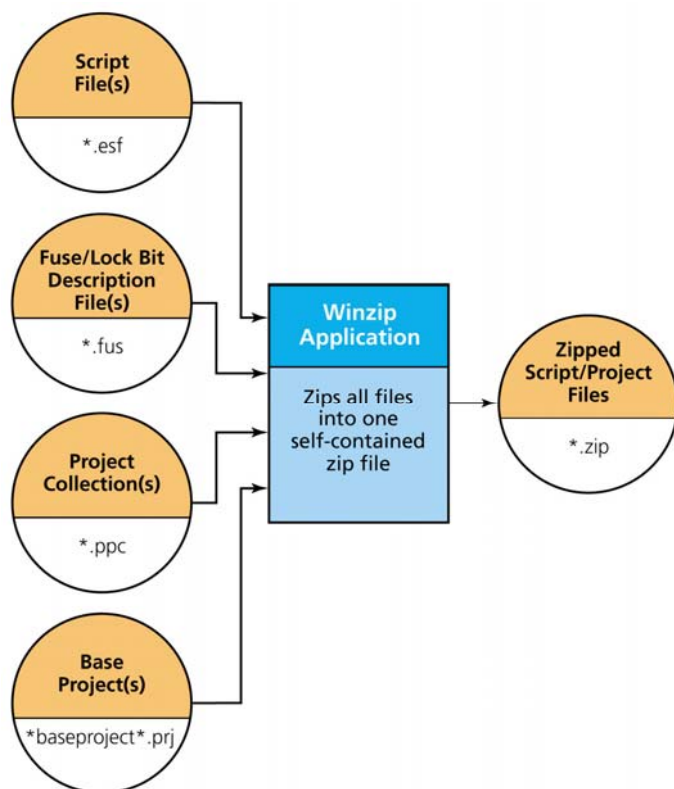
7.4 Distributing scripts and associated files in a ZIP file

It is possible to embed all programmer related files into a single self-contained ZIP file. This approach makes distribution of programming files to multiple programming locations straightforward, as only one file is required. This file can be version controlled and can even be password protected to prevent unauthorised viewing of the files contained within it.

7.4.1 Creating a ZIP file for distribution to a Production Facility

To create a ZIP file containing the Script / Project files which are to be sent to a Production Facility who are running ISP-PRO, please follow the instructions detailed below:

1. Using EQTools, create your Project Files (*.prj), Project Collection File (*.ppc) and Script Files (*.esf).
2. Zip all programmer related files required for ISP-PRO (*.esf, *.prj – base project only, *.ppc, *.fus) into a single ZIP file using the Winzip application (not supplied) – see figure below.



Remember, you should only include the files required by ISP-PRO – see section 6.8.3. Do NOT send the script/project source files to your Production Facility unless you want them to be able to make alterations to the files.

3. Send this *.zip file to your 'Production Facility' via e-mail or disk and forward them the instructions detailed in section 6.8.6.2 of the ISPpro Manual.

Please note: The Interface Database (*.mdb) should NOT be included in this zip file.

7.5 Testing a script file

The Script Debugger can and should be used to test a script and its projects in a production like environment. The Script Debugger uses the same script engine as ISPpro, but gives you additional features such as interrupting and resuming a script.

To test your target device in the script debugger follow the instructions detailed in the table below.

#	Equinox Utility	Action
1	Project Builder	Create a new Project for the target device
2	""	Provide hex files, SPI settings
3	""	Save and Compile the project
4	Project Manager	Create a new Project Collection
5	""	Add the previously created project
6	""	Upload the project collection to your programmer
7	Script Wizard	Create a new script using the Script Wizard
8	""	Select the project as Base Project and AutoProgram1 (You will have to select project inside the project collection)
9	""	Save and Compile the script
10	""	The compiled script file is displayed, select <Run> to run the Script Debugger
11	Script Debugger	Select <Run> to see the script working. Watch the status bar for the current programming status and error messages. Select <Pause> and <Step> to debug the script.