



Oxide User Guide

UG-0050

April 28 2009

Website

www.plxtech.com

Technical Support

www.plxtech.com/support

© PLX Technology, Inc. 2009. All Rights Reserved. The information in this document is proprietary and confidential to PLX Technology. No part of this document may be reproduced in any form or by any means or used to make any derivative work (such as translation, transformation, or adaptation) without written permission from PLX Technology.

PLX Technology provides this documentation without warranty, term or condition of any kind, either express or implied, including, but not limited to, express and implied warranties of merchantability, fitness for a particular purpose, and non-infringement. While the information contained herein is believed to be accurate, no representations or warranties of accuracy or completeness are made. In no event will PLX Technology be liable for damages arising directly or indirectly from any use of or reliance upon the information contained in this document. PLX Technology may make improvements or changes in the product(s) and/or the program(s) described in this documentation at any time.

PLX Technology retains the right to make changes to this product at any time, without notice. Products may have minor variations to this publication, known as errata. PLX Technology assumes no liability whatsoever, including infringement of any patent or copyright, for sale and use of PLX Technology products.

PLX Technology and the PLX logo are registered trademarks of PLX Technology, Inc.

All product names are trademarks, registered trademarks, or servicemarks of their respective owners.

Document number: UG-0050



Prefacevii

- Reference Documentsvii
- Typographic Conventionsviii
- Revision Informationviii

Chapter 1 Introducing and Setting up Oxide

- Installing Oxide2
- Running Oxide and Launching a Utility3
- Initial Setup5
- Configuring Oxide Updates5
- Company Details6
- Setting up Digital Signing6

Chapter 2 Evaluation Workflow

- Before You Start14
- Setting your Target Operating System15
- Identifying Drivers15
- Preparing Your System16
- Editing EEPROM Contents17
 - Using the EEPROM Image18
- Updating Installation Details18
 - Managing Device Details19
 - Specifying the Supported UART Devices21
 - Saving the Driver Package21
 - Installing the Driver Package22
- Branding and Customizing the Driver22
 - Customizing the User Interface22
 - Configuring the Hardware25
 - Restrictions26
 - FIFO Settings27
 - Branding File Completion27
- Preparing the Final Driver Package27
 - Creating the Driver Package28
 - Signing the Driver Package28
- Creating a Driver Installer28
 - Testing and Saving the Installer32

Chapter 3 Design Base

Getting Started	35
Initial View When Installed from the PLX Technology Website	35
Initial View When Installed Using an Oxide CD	37
Design Base Tool Bar	38
Design Base Menu Bar	39
Configuration Options	40
Viewing Content	41
Accessing Content	42
Handling Zip Files	43
Deleting Faulty Files	43
Viewing the File History	43
Accessing the PLX Technology Website	43
Configuring and Downloading Updates	44
Configuring Design Base Updates	44
Downloading Updates	44
Viewing the Update History	46

Chapter 4 EEPROM Editing

Programming EEPROMs with the EEPROM Editor	48
Starting the EEPROM Editor	48
Golden Image Files	49
Using the EEPROM Editor	50
EEPROM Editor Tool Bar	51
Screen Icons	52
EEPROM Editor Menu Bar	52
EEPROM Manipulation	53
Selecting a Device	53
Selecting an EEPROM	54
Device Configuration Settings	54
Zone Settings	55
Editing EEPROM Settings	56
Saving an Image File	57
Programming an EEPROM	57

Chapter 5 INF Editing

Managing .inf Files	59
.inf File Additional Information	59
Starting the INF Editor	60
Using the INF Editor	61
INF Editor Tool Bar	62
INF Editor Menu Bar	62
File Pane Contents	63
Information Pane Contents	64
INF File Manipulation	66
Opening a Package	66
Creating a Package	67
Saving a Package	67
Digitally Signing a Driver Package	67
Creating a Final Driver Package	68

Chapter 6 Driver Branding

Branding Files	69
Starting the Driver Branding Utility	70
Branding Tool Bar	72
Branding Menu Bar	73
Tab Hierarchy Contents	74
Information Pane Contents	74
Opening a Branding File	76
Creating a Branding File	76
Opening and Modifying an Installed Branding File	76

Chapter 7 Installer Creation

Starting Installer Creation	78
Installer Creation Tool Bar	78
Installer Creation Menu Bar	79
File Signing	80
Setting up Installer Creation	80
Defining Installer Text	82
Specifying Installer Graphics	85
Specifying Driver Files	89
Saving an Installer	90
Testing an Installer	90
Command Line Parameters	90

Chapter 8 OxShell

Running OxShell	93
Using the Command Shell Tool	94
Viewing and Printing Command Shell Help	95
Closing OxShell	95
Using the Command Line Tool	95
Viewing Command Line Help	96
Commands	96
Editing Keys	98
Error Messages	99
Using OxShell with the Oxide EEPROM Editing Utility	100



The *Oxide User Guide* explains how to use Oxide, an PLX Technology Windows® toolkit. Oxide comprises utilities that can be launched independently or through the launcher. Oxide facilitates programming the EEPROM on PLX Technology PC bridge and UART devices, and allows you to customize drivers and user interfaces. It simplifies customizing driver packages for third party solutions. This guide is relevant to hardware and software developers working on implementations using those products.

This guide assumes that you understand the capabilities of PLX Technology drivers, bridge and UART products and are familiar with PCI, PCI Express®, PC Card and compact flash interfaces, and the component parts of Windows device drivers, such as **.inf** files.



This user guide refers to Oxide V3.0 or later.


Reference Documents

The following documents are referenced in this guide:

<i>PCI Local Bus Specification</i>	Revision 2.3, March 29, 2002
	PCI Special Interest group
PC Card Standard	Revision 8.0, PCMCIA/JEITA, 4-2001
PCI Express Base Specification	Revision 2.0, PCI SIG, December 20, 2006

Typographic Conventions

In this document, the following conventions apply.

Convention	Meaning
<i>Italic Letters With Initial Capital Letters</i>	A cross-reference to another publication
Title	A cross-reference to another section within the document
1, 2, 3	A numbered list where the order of list items is significant
■	A list where the order of items is not significant
	Significant additional information
Courier	Software code
Bold	Significant names, for example of files, fields, buttons, menus and options Text you type
Menu - Option	Go to the Menu menu and select the Option option
<text in italics>	Variable information or values that you enter or that are displayed

Revision Information

The following table documents the revisions of this guide.

Revision	Modification
April 28 2009	Rebranded and reformatted
December 03 2008	Restructure to improve information flow, new functionality and OxShell information added
March 2008	Modifications to document improvements and increased functionality
November 2007	Modifications to document improvements and increased functionality
September 2007	First publication

Introducing and Setting up Oxide

Oxide is an integrated package of utilities that can be launched and used independently or through the Oxide launcher.

The Oxide utilities are:

- Design Base—providing access to the latest PLX Technology technical documents, software and support information. For more information, see Chapter 3 [Design Base](#)
- EEPROM editor—modifying the EEPROM on PLX Technology PC bridge and UART devices. For more information, see [Editing EEPROM Contents](#) on page 17 and Chapter 4 [EEPROM Editing](#)
- INF editor—customizing device driver packages. For more information, see [Updating Installation Details](#) on page 18 and Chapter 5 [INF Editing](#)
- Branding—customizing user interfaces and driver functionality. For more information, see [Branding and Customizing the Driver](#) on page 22 and Chapter 6 [Driver Branding](#)
- Installer Creation—creating an installer package for PLX Technology Windows device drivers. For more information, see [Creating a Driver Installer](#) on page 28 and Chapter 7 [Installer Creation](#)

The Oxide toolkit also includes the OxShell utility, providing a command line and command shell tool for advanced EEPROM programming.

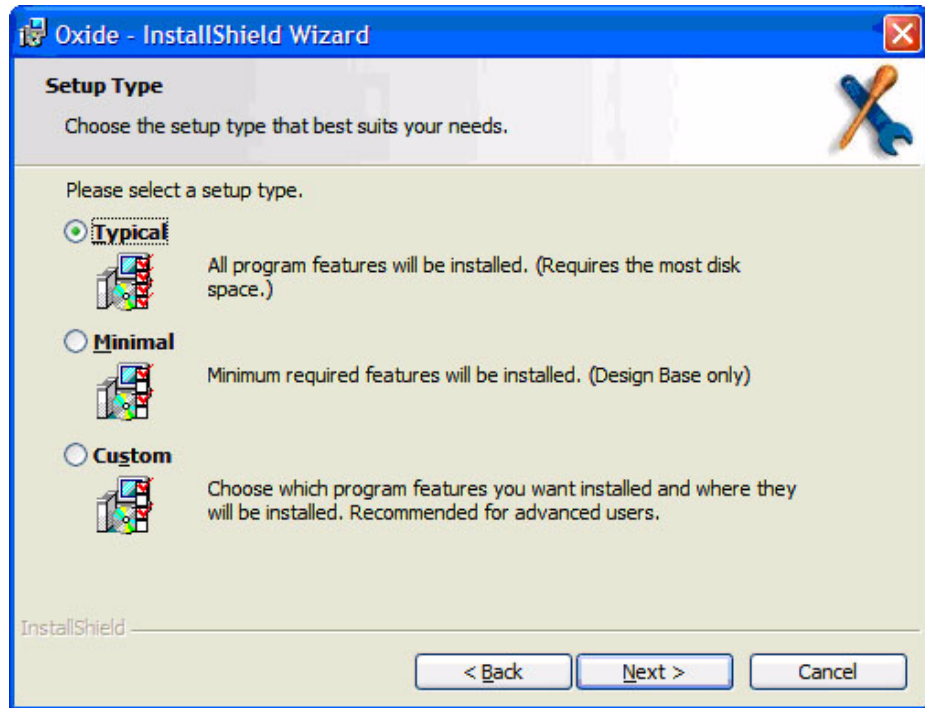
By combining the utilities into a single tool, Oxide:

- Provides all relevant information for supported PLX Technology devices, ensuring that all required documentation, schematics and software are readily accessible and up to date
- Streamlines the process of EEPROM programming, updating installation files to match and tailoring the user interface to deliver a consistent device package
- Imposes consistency on specialized files, allowing you to specify information once, and automatically replicating it throughout the driver package as necessary
- Removes the need for specialist knowledge and access to source code when customizing and branding a driver for end user solutions

Oxide supports a variety of PLX Technology devices and can run under Windows 2000, XP and Vista (32 bit only).

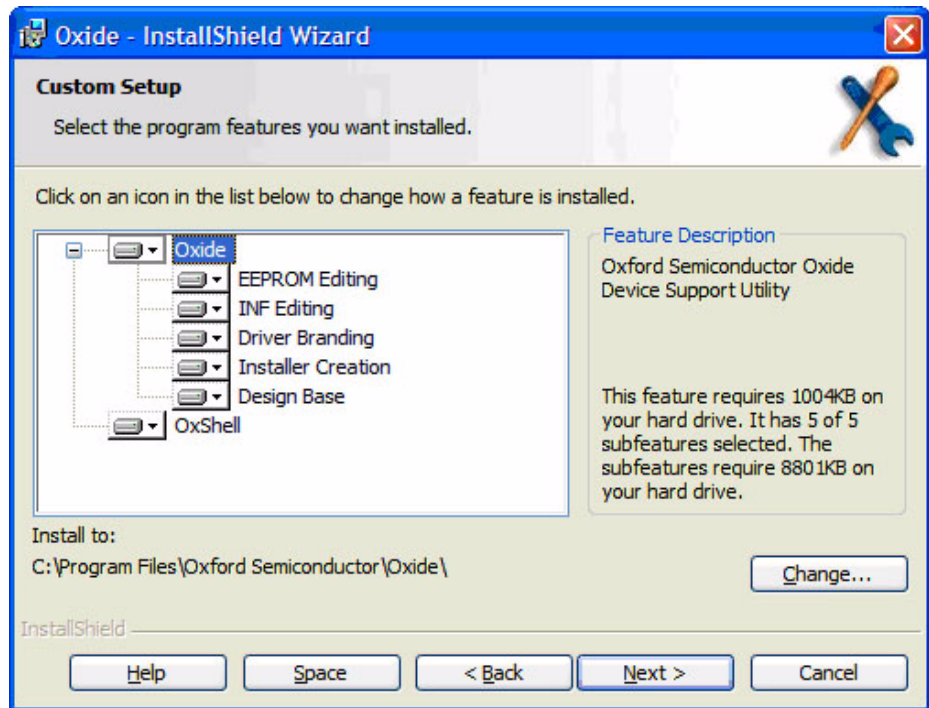
Installing Oxide

Oxide is supplied as a self-installing executable file. To install Oxide, run the file, supplying registration details as required to obtain the Setup Type window shown below.



For a typical setup, the installation process installs all Oxide utilities, including OxShell. However, it is possible to install only selected components of the Oxide toolkit:

- To install only the Design Base, click **Minimal**.
For more information on the Design Base, see Chapter 3 [Design Base](#)
- To choose other utilities, click **Custom**. This opens the Custom Setup window, shown below



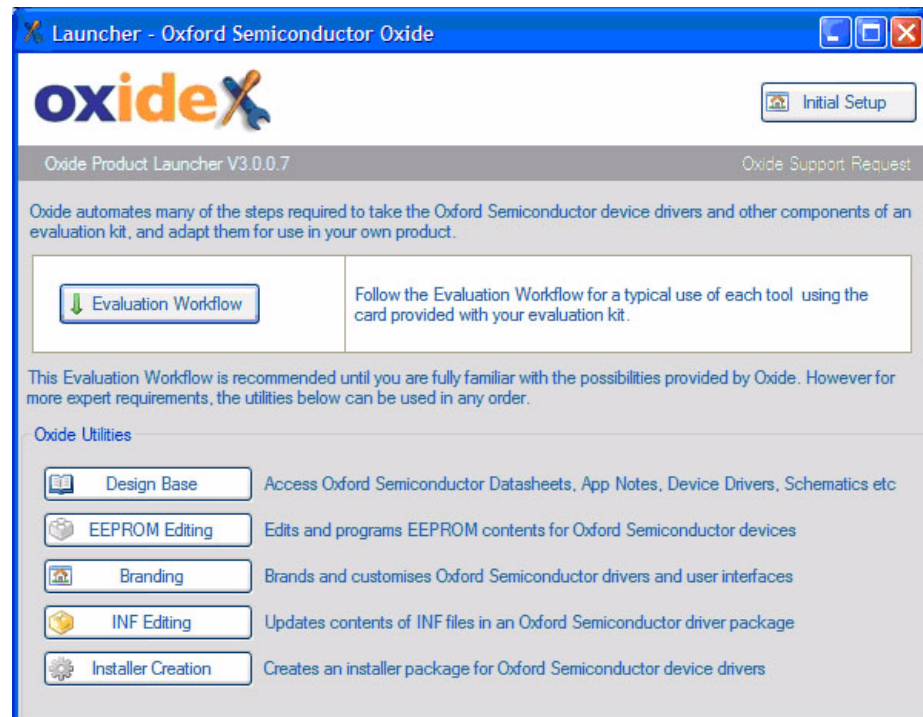
This user guide assumes that you undertake a **Typical** installation.

The Oxide installation process places an Oxide icon on your desktop, and inserts shortcuts into the Windows Start menu for each utility.

Running Oxide and Launching a Utility

When you run Oxide, it automatically checks the PLX Technology website for newer versions of the toolkit. If a later version exists, Oxide asks you for permission to download it or you can choose to defer the download. When you download a later version, you can either save it for later installation or install it immediately, as described in [Installing Oxide](#) on page 2.

To open the utility you require, go to the Windows Start menu and select the relevant shortcut. Alternatively, you can run Oxide by clicking the Oxide icon on your desktop, which opens the Launcher window.

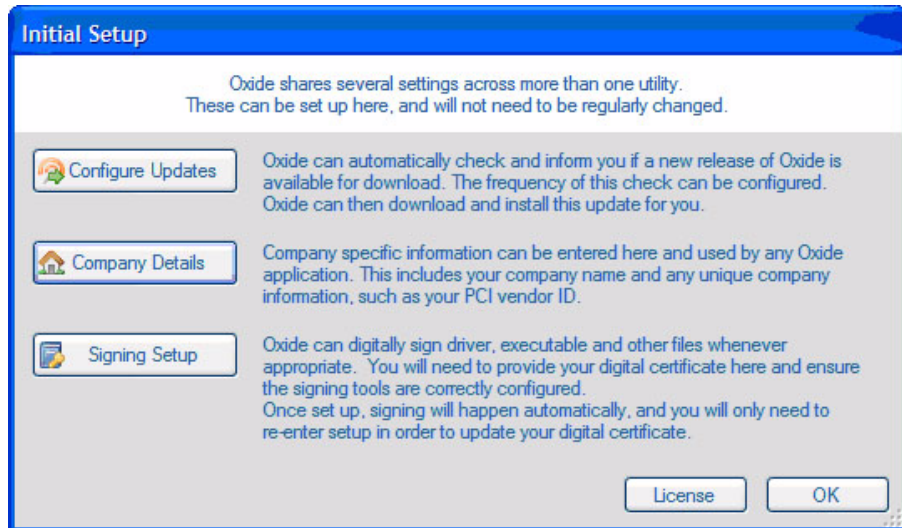


The Oxide Launcher provides the following options:

- **Initial Setup**—settings to define after installation and before using any Oxide utilities. You can change these settings at any time, but they do not need to be changed often. For more information, see [Initial Setup](#) below
- **Evaluation Workflow**—a step by step example of how to configure, customize and package device drivers into a deliverable solution. For more information, see Chapter 2 [Evaluation Workflow](#)
- **Oxide Utilities**—to run a specific Oxide utility. Shortcuts for each utility are also available in the **Start** menu. For more information on a specific utility, see the relevant chapter in this guide

Initial Setup

You can configure options that are shared by more than one Oxide application. It is recommended that you configure these before any serious work or evaluation is performed using Oxide. You can change the values at any time subsequently, but typically you do not change them often. To do so, on the Launcher window, click **Initial Setup** to open the Initial Setup window.

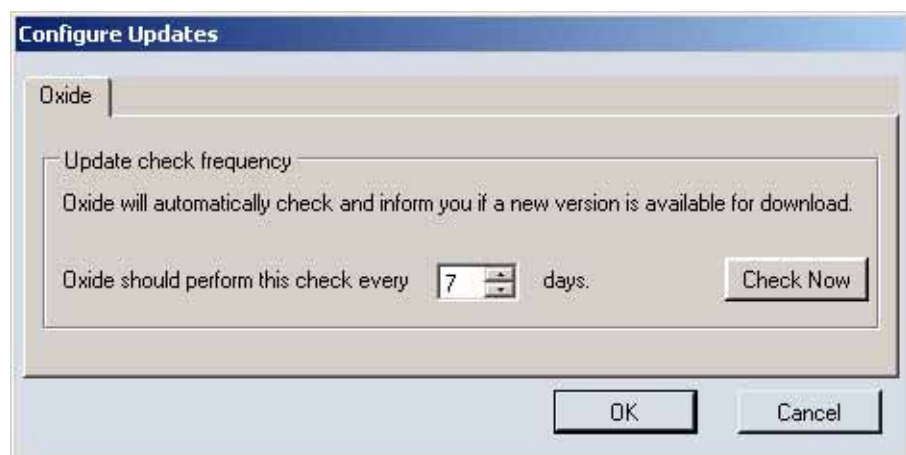


The options are:

- **Configure Updates**—see [Configuring Oxide Updates](#)
- **Company Details**—see [Company Details](#) on page 6
- **Signing Setup**—see [Setting up Digital Signing](#) on page 6

Configuring Oxide Updates

To specify the frequency for checking the PLX Technology website for Oxide updates, on the Initial Setup window, click **Configure Updates**. The Configure Updates window opens.



You can:

- Set the checking interval in days
- Check for updates immediately. If an update is available, you can download and install it

Company Details

To ensure consistency across EEPROM specifications, driver packages and user interfaces, you use the Company Details window to specify company and ID information once. The information is used in and can be accessed from the EEPROM editor and the INF editor.

To enter company details, on the Initial Setup window, click **Company Details**. The Company Details window is shown.

PLX Technology devices are supplied with vendor ID 1415, but you must update the vendor and manufacturer IDs to identify your own devices. The EEPROM editor displays warnings if you do not modify the IDs.



To check IDs on either the PCMCIA or PCI website, click .

Setting up Digital Signing

Windows files such as executables, DLLs, driver files and catalog files are usually digitally signed before redistribution. This is particularly important for files contained in a Windows driver package, and is a more stringent requirement in Windows Vista.

Using the Signing Setup window, you can set Oxide to automatically digitally sign these files when Oxide generates or modifies them.

Typical uses of digital signing in Oxide are as follows:

- **Inf Editing**—driver package automatically creates and digitally signs catalog files, and signs SYS and DLL files in the driver package
- **Installer Creation**—self extracting executables, plus the setup.exe in the installer are digitally signed

Before you can configure Oxide for digital signing, you must ensure that:

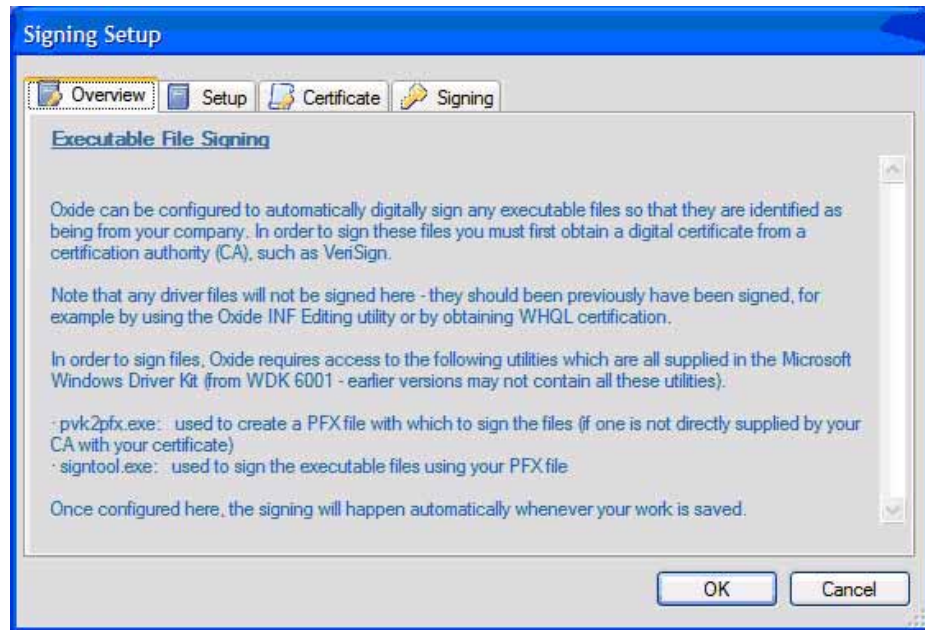
- You obtain a digital certificate from a certification authority (CA) such as VeriSign. This identifies your package as having been created by your company. If you do not have a certificate, you can still configure Oxide to create valid catalog files (see Chapter 5 [INF Editing](#)), but it cannot digitally sign them. Digitally signing a driver using your own certificate is not the same as WHQL certification: you do not need to pass any WHQL tests in order to sign your own driver package
- The Windows Driver Kit (WDK) is installed on the same PC as Oxide. Several utilities are required which are supplied as part of WDK. It is recommended that only WDK version 6001 or higher is used, as not all utilities may be present in earlier WDK versions

The WDK utilities you use are:

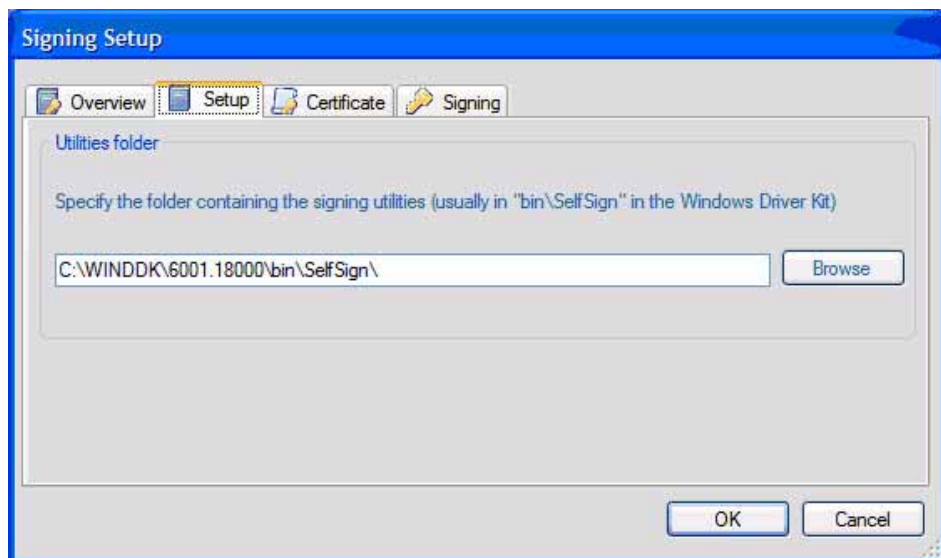
- **pvk2pfx.exe**—used during signing setup, or when you update your digital signature certificate, to create a PFX file from your digital certificate (if one is not directly supplied by your CA). The PFX file is used to sign the files
- **signtool.exe**— used by any Oxide utility, to perform the digital signing of files
- **inf2cat.exe**—used by the Inf editor, to recreate valid catalog files when saving a driver package in the INF editor. This does not need a digital certificate, but must be performed for the driver package to be digitally signed.
Earlier WDK versions (such as 6000) do not have inf2cat.exe. You can still use an earlier WDK version, but you must manually copy the inf2cat utility into the same folder as the other two utilities (typically in the **bin\SelfSign** folder). Inf2cat.exe was previously also provided as part of the Microsoft Winqual Submission kit. As well as the inf2cat.exe you must copy all the DLL files beginning with **Microsoft.Whos**

To set up automatic digital signing:

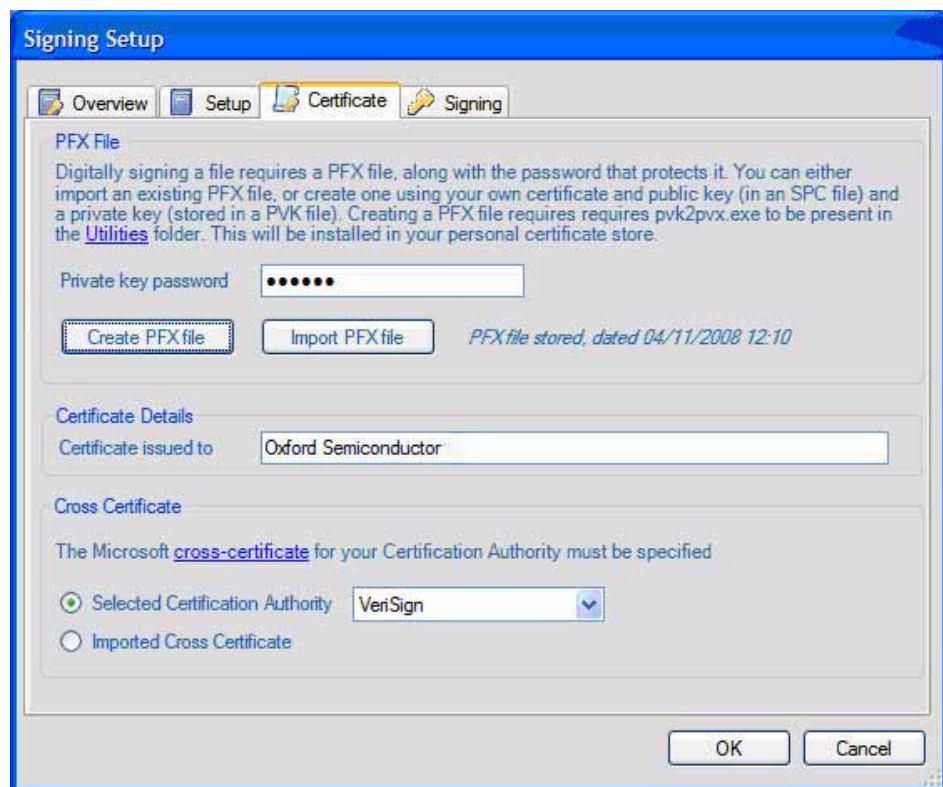
- 1 On the Launcher window, click **Initial Setup**. The Initial Setup window is shown.
- 2 Click **Signing setup**. The Signing Setup window is shown, with the Overview tab active.



- 3 Click the Setup tab to display it.



- 4 Oxide tries to locate a suitable WDK on your PC and shows the path to the folder containing the required utilities. If it is unable to do so, or you want to provide an different path, you may enter a new path here. You must enter a path to a folder containing all the required utilities, and not the path to the base of the WDK. Typically these utilities can be found in a **bin\SelfSign** folder.
- 5 Click the Certificate tab to display it.



- 6 To digitally sign your files, Oxide needs your certificate in the form of a PFX (personal information exchange) file, plus the password that protects your private key. Do one of the following:
 - If your certification authority (CA) has provided a PFX file, enter the password and click **Import PFX file**.
 - If your CA has provided your certificate in two files—a certificate and public key stored in an .spc file, and a private key stored in a .pvk file (along with a password that protects it)—enter the private key password and click **Create PFX file**. When prompted, specify your .spc and .pvk files. The pvk2pvk.exe utility creates a PFX file for Oxide to use.
- 7 The PFX file must be imported into the certificate store on your PC: a wizard window similar to the following is displayed.

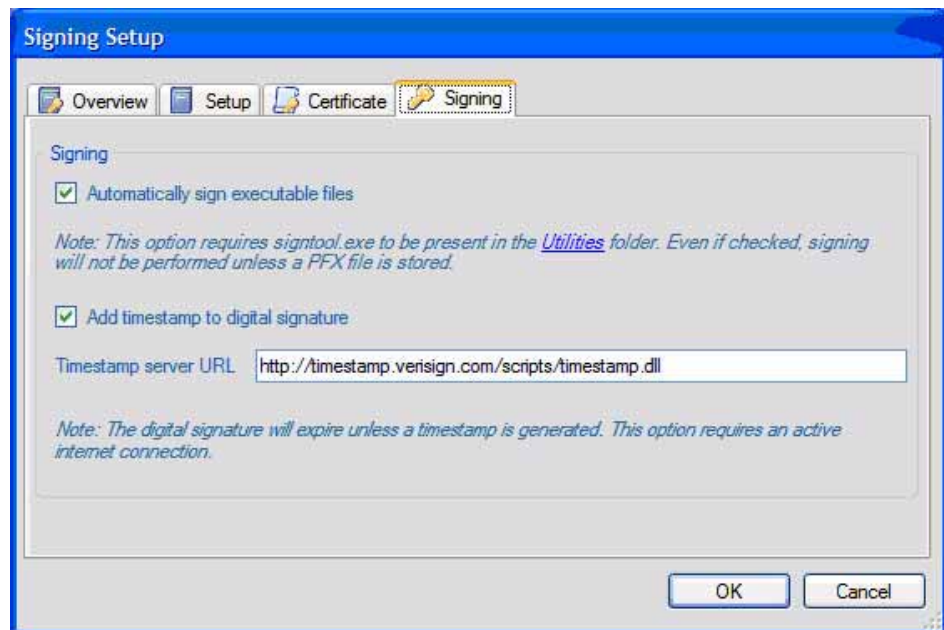


- 8 Click **Next** and work through the wizard. Usually you do not need to change the default values displayed in subsequent wizard windows. However when prompted, you must re-enter the same private key password that you entered in the Certificate tab.
- 9 Oxide attempts to identify the company name and Certification Authority (CA) that issued your certificate, and displays this information at the bottom of the Certificate tab.



- 10 Check that this information is correct, otherwise your files cannot be signed correctly. Oxide uses this information to select the correct cross certificate for your CA. If your CA is not supported directly by Oxide, you must import the appropriate cross certificate file (*.cer) by clicking **Imported Cross Certificate**.

- 11 To enable digital signing, click the Signing tab to display it.



- 12 To enable digital signing, select **Automatically sign executable files**.
- 13 To add a timestamp to a digital signature, select **Add timestamp to digital signature**. If it is not added, your files are still signed, but the signature expires when your certificate expires. If a timestamp is added, then the signature never expires. When you add the digital signature with a timestamp, you must have an active internet connection and access to a timestamp server. You can specify an alternative timestamp server to be used if required. Oxide defaults to using the VeriSign server.
- 14 Click **OK** to save your changes.

Digital signing now happens automatically whenever required.

This page is intentionally blank



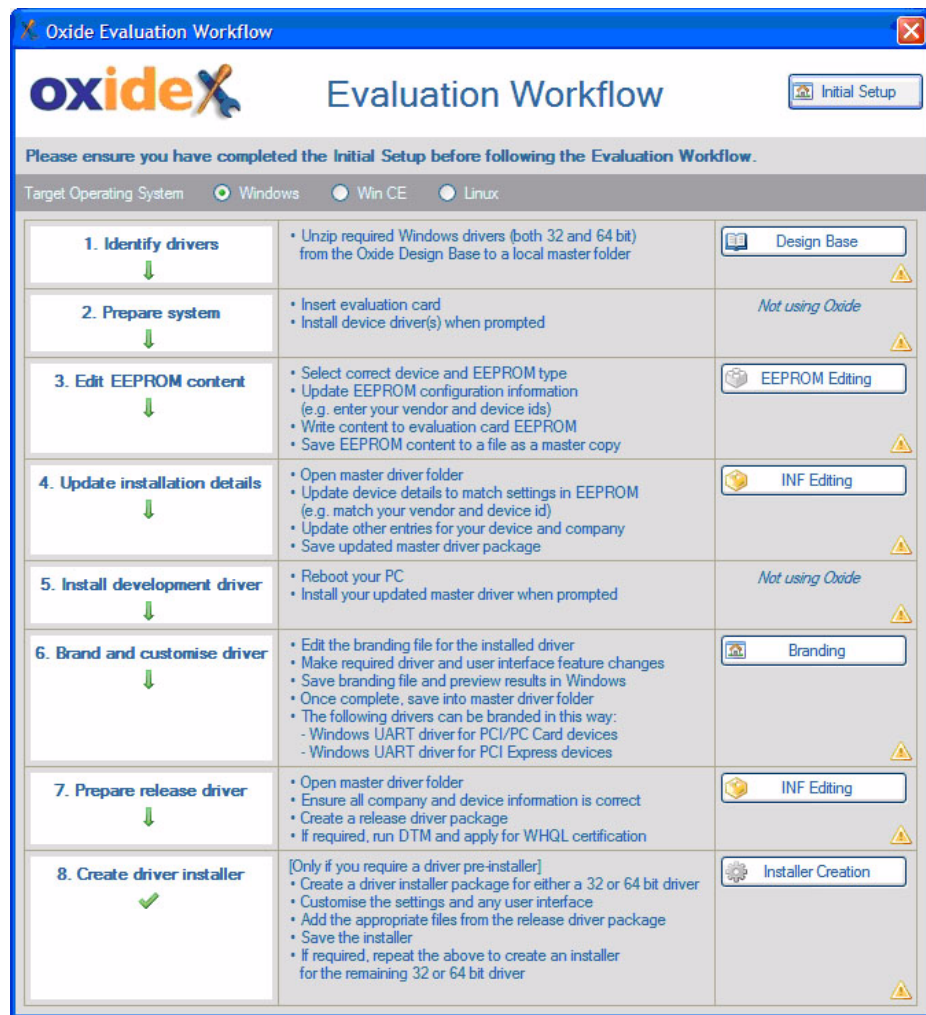
Evaluation Workflow

Oxide integrates several powerful utilities into one toolkit to facilitate the creation of your company's product from PLX Technology device and driver software. For the majority of applications, you can use Oxide to brand, customize and target your product at your application. You do not need specialist driver writing skills or access to the original source code.

Although each utility is powerful in its own right, their true capabilities are unlocked when used together to create your solution.

This chapter describes a step-by-step example of using the Oxide utilities to prepare a configured board and driver package, modifying the PLX Technology UART driver for a PCI Express device.

To begin, go to the Oxide Launcher window (see [Running Oxide and Launching a Utility](#) on page 3) and click **Evaluation Workflow**. The Evaluation Workflow window is shown.



You can use the Oxide tools in any order you want. However, it is recommended that you follow the example workflow to familiarize yourself with Oxide. You can use this workflow with an evaluation board from an PLX Technology evaluation kit.

Before You Start

Before starting the evaluation workflow, you must complete the initial setup steps, and you must have all the latest Oxide updates. For more information, see [Initial Setup](#) on page 5.

You can access the Initial Setup window using the Oxide Launcher window, or by clicking Initial Setup on the **Evaluation Workflow** window shown above.


You must:

- Ensure you are running the latest version of Oxide. To do so, on the Initial Setup window, click **Configure Updates**; then on the Configure Updates window, click **Check Now**. For more information, see [Configuring Oxide Updates](#) on page 5
- Ensure you have entered your company details, including vendor and manufacturer IDs. To do so, on the Initial Setup window, click **Company Details**; then on the Company Details window, enter the details. For more information, see [Company Details](#) on page 6
- Ensure you have configured digital signing and imported your digital certificate, if you want automatic signing. To do so, on the Initial Setup window, click **Signing Setup**; then follow the instructions in [Setting up Digital Signing](#) on page 6

Setting your Target Operating System

Oxide provides different levels of support for different operating systems. To ensure the evaluation workflow provides the correct feedback in its step-by-step guide, specify the operating system you are working with: go to the top of the Evaluation Workflow window and click the target operating system you want.

For this evaluation, click **Windows**. Information on the tasks to be performed at each stage is displayed.

If at any stage you are obtaining unexpected results, click  **Having Problems?** for some useful hints and tips.

Identifying Drivers

The Oxide Design Base is a repository of collateral supporting a range of PLX Technology devices, including device drivers, technical documents, schematics and other support files. For more information about the Design Base, see Chapter 3 [Design Base](#).

To extract the drivers required for your evaluation:

- 1 To find out which drivers you need, see the getting started guide for the evaluation board. If you need a copy of the getting started guide:
 - On the Evaluation Workflow window, click **Design Base**. The Design Base window is shown.



- In the left hand pane, select the device being evaluated (in this example the OXPCle952). All files contained in the Design Base that are relevant to this device are displayed.
 - Go to the top of the right hand pane and double-click the PDF format file to view it.
- 2 Scroll down the right hand pane until you find the required device driver(s). For this example, the **PCI Express Native UART Driver for Windows 2000, XP & Vista**.
 - 3 Right-click the zip file containing the driver and select **Unzip To - Desktop**. The device driver is extracted and placed in a subfolder under a PLX Technology folder on your desktop, and opened in Windows Explorer.

Preparing Your System

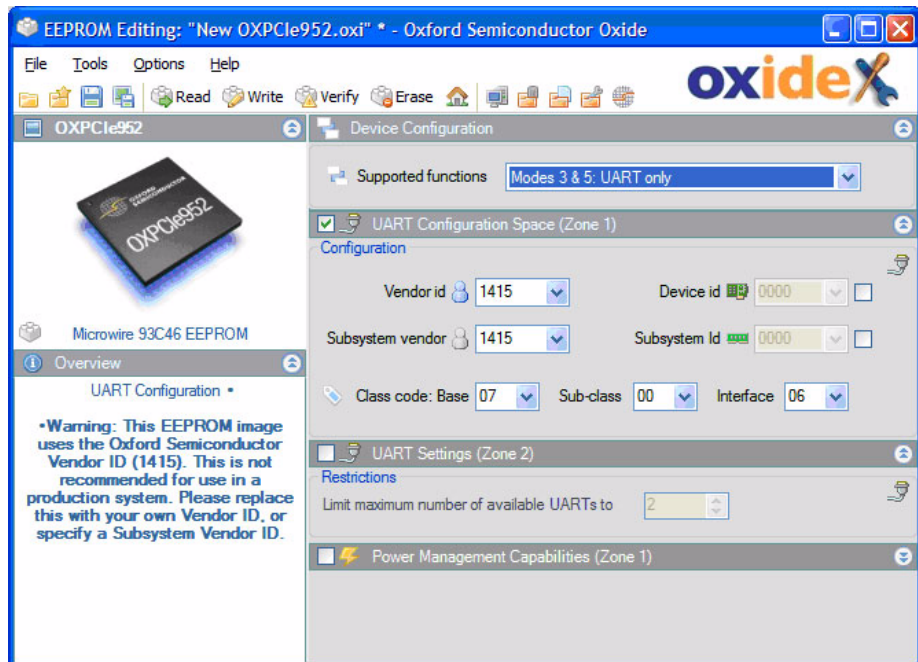
Install the evaluation board in your PC, and install the device drivers for the evaluation board. For more information, see the getting started guide for your device.

Editing EEPROM Contents

You must program the EEPROM contents on your board to identify your company and you must customize other device-specific settings. You do this using the EEPROM editor. For more information on the EEPROM editor, see Chapter 4 [EEPROM Editing](#).

To edit EEPROM contents:

- 1 On the Evaluation Workflow window, click **EEPROM Editing**. The EEPROM Editing window is shown.




- 2 If the window shows a device other than the one you want, either click the device image or use **Options - Select Device**.
- 3 Specify the correct EEPROM type by clicking the device image and choosing the EEPROM type you need.
- 4 Select the device type. The EEPROM editor defaults to this for future startups until you change the device type again.
- 5 The EEPROM Editing window shows values for a typical golden EEPROM image, from which you can start creating your own.
- 6 If you already have an EEPROM image you want to use, load it by clicking **Open Image**.
- 7 If your device has a valid EEPROM image, click **Read EEPROM** to use those EEPROM contents.
- 8 To assign your vendor ID to any function in your device, click **Company Details** in the tool bar. The fields on the window are populated with the appropriate values.

- 9 To generate the correct EEPROM contents, some PCI Express devices must be aware of the device configuration used on your board. In this case, ensure you select the correct value in **Device Configuration - Supported Functions**. The screen sample above shows the OXPCle952 configured in either mode 3 or mode 5, with the UART function only enabled.


You may have to complete other fields in the EEPROM Editing window, depending on your device. The window displays the relevant fields.

Using the EEPROM Image

To program your device with the changes, click  **Write EEPROM**. It is also recommended that you save your image to disk as an **.oxi** file. You can reload it and edit it offline without needing access to a device.



Oxide is not a production programming tool; however, you can use the **.oxi** file in a production environment if it is required by the PLX Technology OxShell utility. You can also save the EEPROM contents in a format used by gang

programmers, such as Motorola S-Record or Intel Hex format. To do so, click  **Save As** and select the appropriate file format.

Having changed the EEPROM contents, if you restart your PC, your device is no longer recognized by the supplied PLX Technology driver package. You now need to use the INF editor to modify the driver package to recognize your device; see [Updating Installation Details](#).

Updating Installation Details

Details used by the operating system to identify devices and install the drivers are stored in **.inf** files and include:

- Vendor and device IDs used to recognize the device
- Names displayed when the driver is installed and when the device is viewed in the Device Manager
- Version and company information, displayed on the device property page
- The driver file names (which must be unique to your company)

You use the INF editor to edit these details, without needing to know anything about **.inf** file formats or the structure of the driver. For more information about the INF editor, see Chapter 5 [INF Editing](#).

The following example describes editing the PLX Technology PCI Express UART driver package.

The example assumes that you have configured Oxide to automatically generate catalog files and digitally sign them. For information on how to do this, see [Initial Setup](#) on page 5.



To prepare both the 32 bit and 64 bit Windows driver, you can update the **.inf** files in both packages simultaneously, by opening a parent folder that contains subfolders for both 32 and 64 bit driver versions. You are presented with the same information to complete, and when you save the package, both 32 bit and 64 bit **.inf** files are updated with the information.

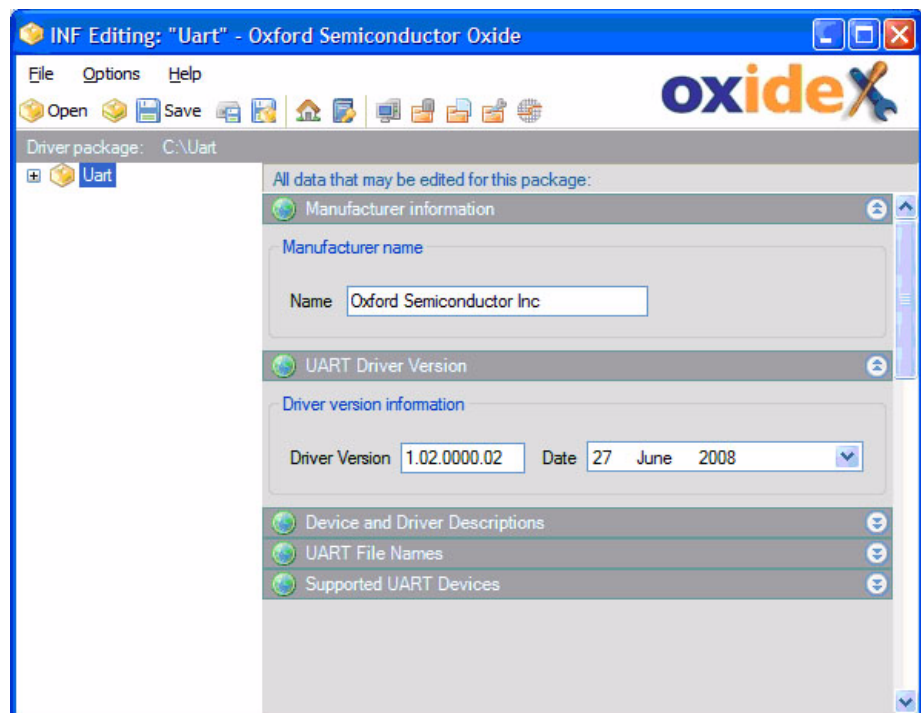
Managing Device Details


You must ensure that the device details in the .inf files match the EEPROM details you entered. To manage device details:

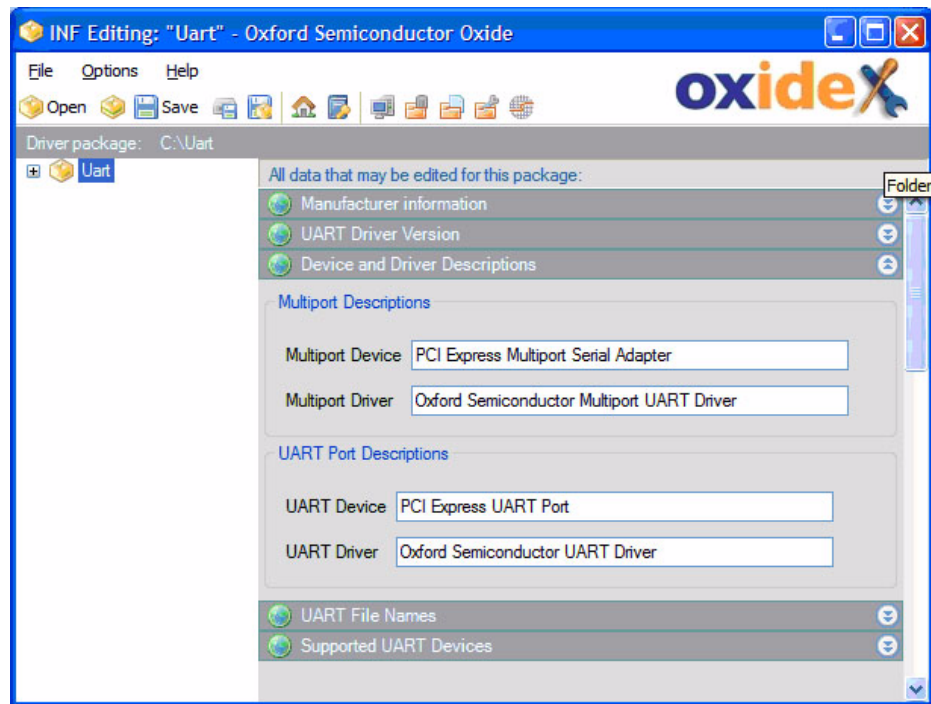
- 1 On the Evaluation Workflow window, click **INF Editing** to display the initial INF Editing window.



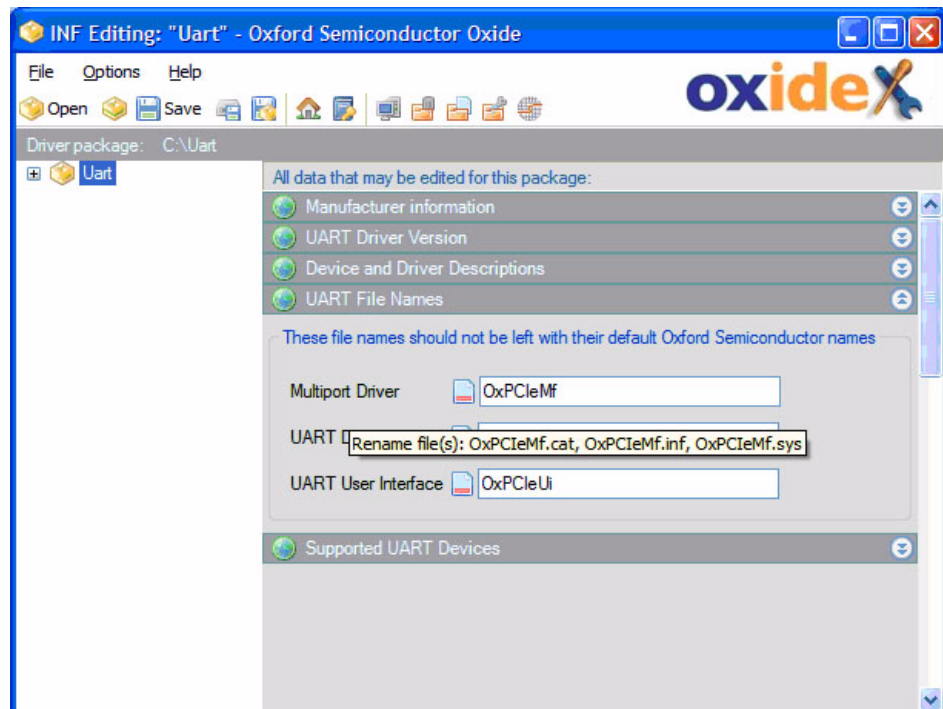
- 2 Click **Open Package** and select the driver package you saved in the PLX Technology folder on your desktop. The INF Editing window is shown.



- 3 To update the manufacturer name and vendor IDs, click  **Company Details**; the fields are populated with the relevant values.
- 4 Under **UART Driver Version**, you can provide your own driver version and date.
- 5 You use other fields to specify the name by which the device is known in the Windows Device Manager, and the description displayed when installing the driver, as shown below.



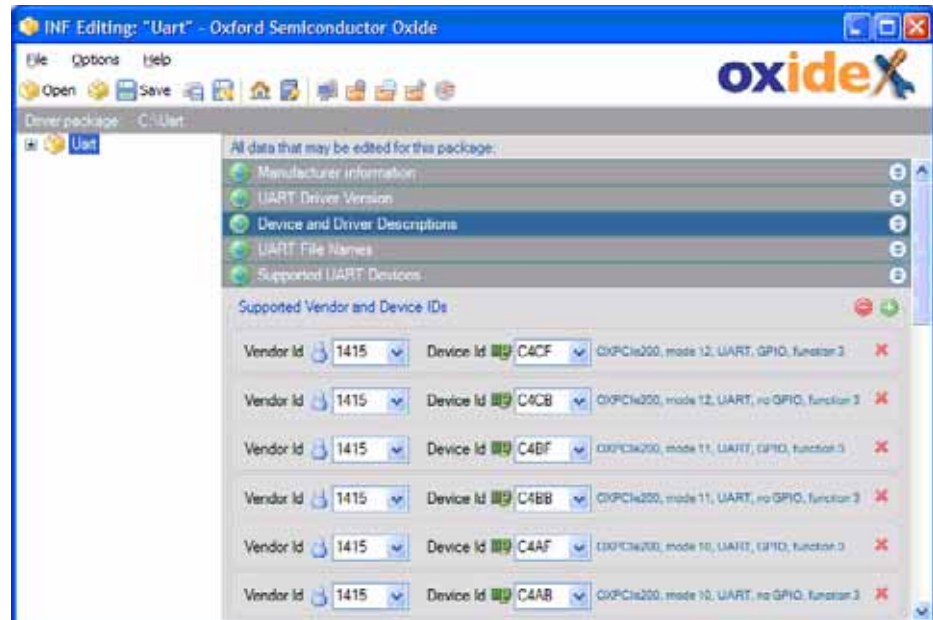
- 6 .Most driver packages allow you to rename the files in the driver package. To minimize the chance of naming clashes with other companies that use PLX Technology devices and drivers, it is strongly recommended that you change the default file names supplied in the PLX Technology driver package. To change file names, click **UART File Names** to show the relevant fields.






- 7 Enter file names that are unique to your package. The files are renamed when you save the package.

Specifying the Supported UART Devices

The Supported UART Devices section specifies the vendor and device IDs recognized by this .inf file, and so devices that are supported by this device driver.



By default all device IDs supported by the supplied PLX Technology driver package are listed, although you probably only need a small number of them. You can click :

-  to delete an entry
-  to add an entry
-  to delete all entries



For PCI Express devices, if you are using one of the default device IDs generated by the device, a description of the device and configured mode that it supports is shown next to the value you entered, to confirm that you have the expected value.

Saving the Driver Package

When you have completed all the required changes, click **Save** to update the .inf files in your driver package with the information you entered, wherever it is used. The same information is often used in multiple places across more than one .inf file. This approach ensures that consistent information is guaranteed across all files.

Installing the Driver Package

Using the driver package stored in the PLX Technology folder on your desktop, you can now reboot your system and install the new driver for your device with its updated EEPROM content. Check the messages displayed during installation and the descriptions displayed in the Windows Device Manager and Device Properties pages to see where the strings you entered are used.

Branding and Customizing the Driver

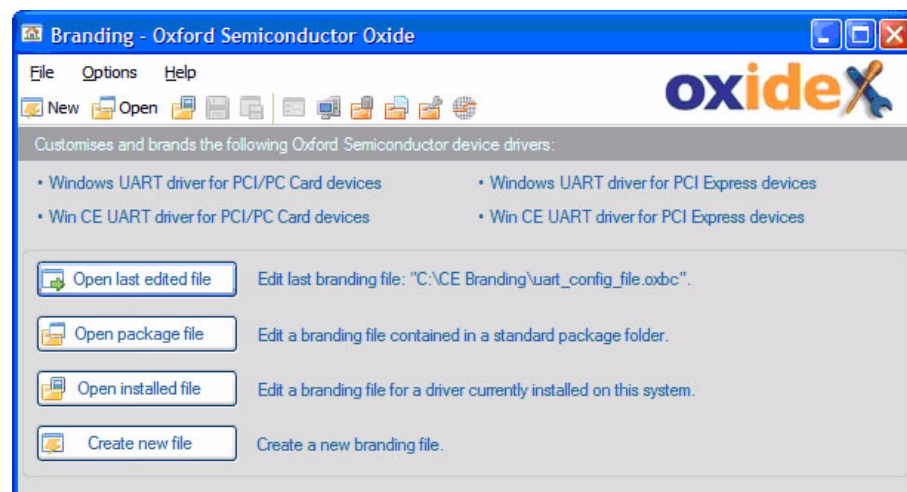
With the driver now installed for the PCI Express UART functionality, you can brand and customize its operation. For more information on branding, see Chapter 6 [Driver Branding](#).



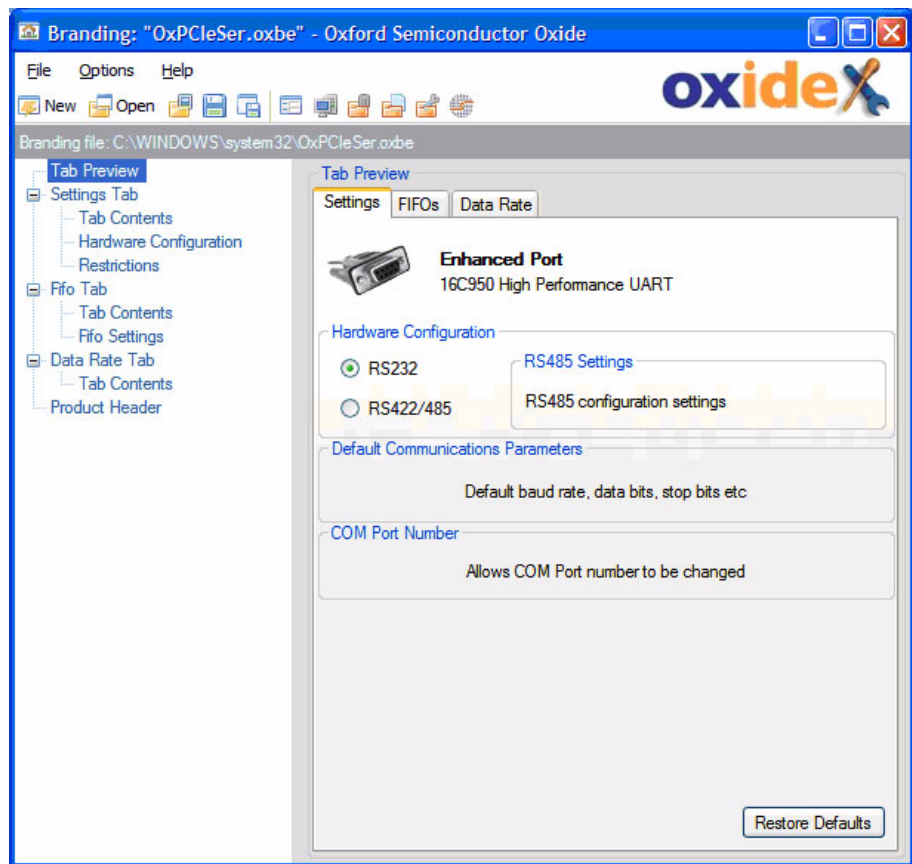
It is not essential to have the driver installed and running on a machine for branding, but it does enable you to see changes to the driver user interface immediately. Alternatively, you can open the branding file in your master driver package.

Customizing the User Interface

- 1 To open the Branding utility, on the Evaluation Workflow window, click **Branding**. The initial Branding window is displayed.




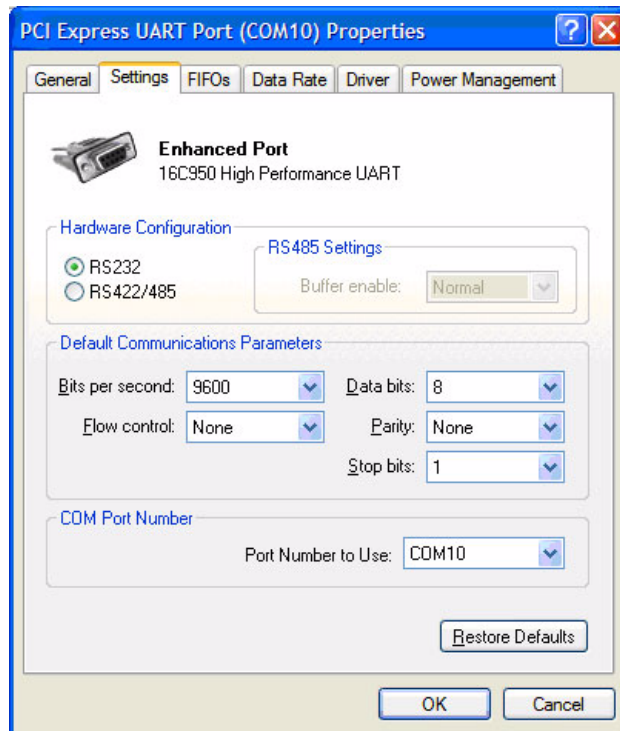
- 2 To open the folder which contains the branding file for the driver you just installed, click **Open installed file**. The file name extension is **.oxbe**.
- 3 Select the branding file and open it. The Branding window is displayed.



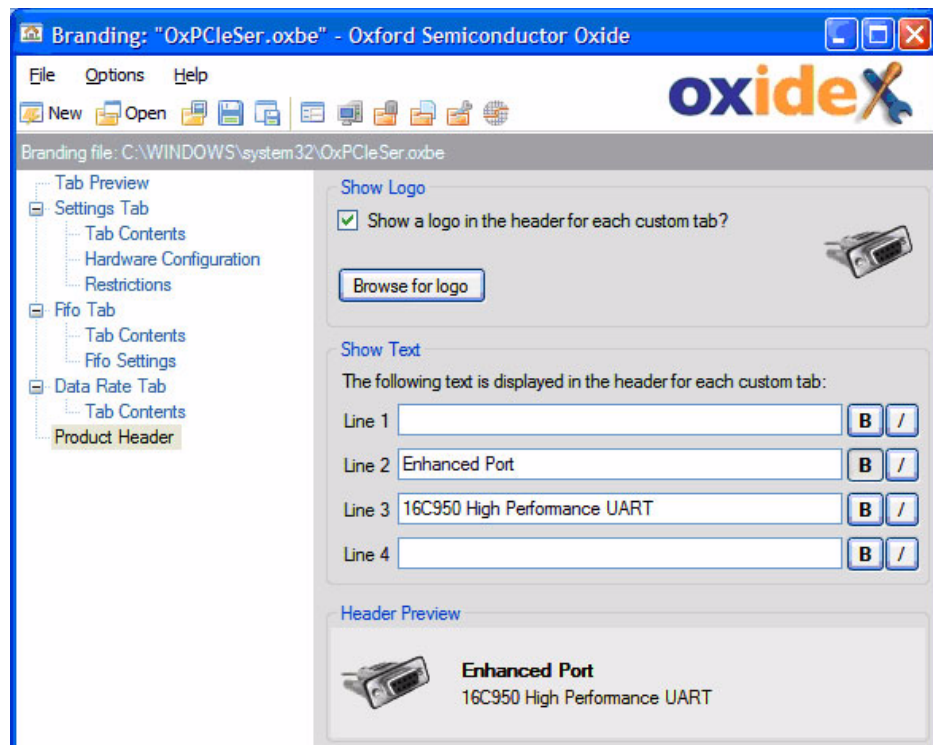
You can customize the user interface by renaming and removing tabs, removing sections of tabs, and by setting up a header for your tabs.

As you make your changes, Oxide gives an approximation of the user interface, showing the additional tab pages added to the standard Windows user interface.

To see the user interface in Windows, save the file and click  **View installed driver's user interface** on the tool bar. An example is shown below, showing the device properties for one of the installed ports using the selected driver.



To display a standard header at the top of each custom tab, you use the Product Header window.



You can use a 64 x 64 pixel full color logo (with transparency) at the left of this area. It is automatically resized to fit the available area, retaining the correct aspect ratio of the original image. The following file types are acceptable:

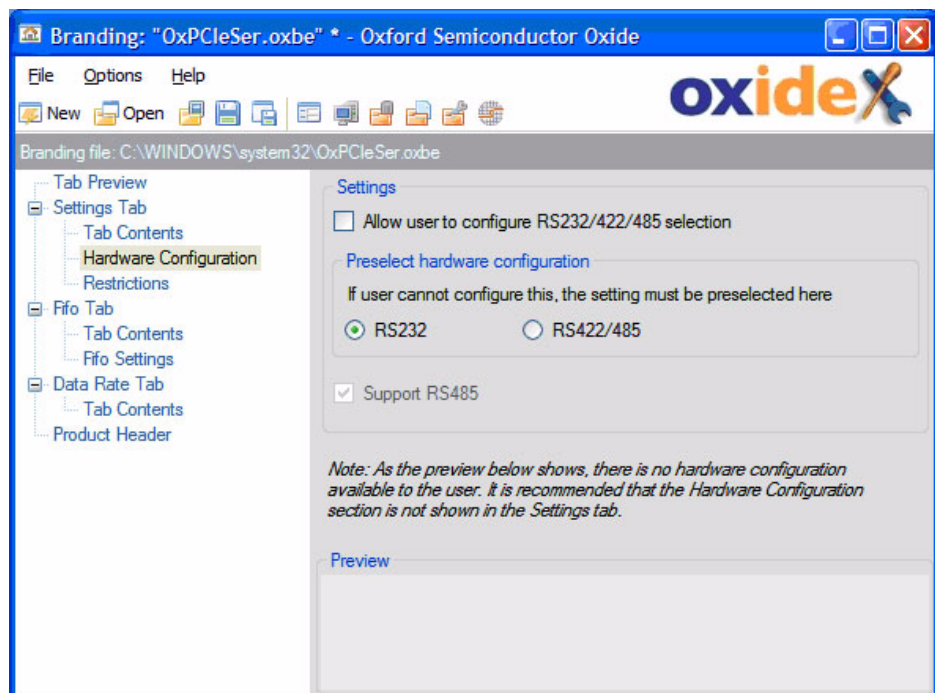
bmp, gif, ico, jpg, png, tif, wmf

You can include up to four lines of text, with information such as the company name or device name. Each line can independently be emboldened or italicized.

For the PCI Express UART driver, you can configure some more advanced features, which affect the operation of the driver and not just the appearance of its user interface. If you configure these, the changes may not become active until the driver is next restarted (at system reboot or after disabling and re-enabling the device using Device Manager).

Configuring the Hardware

You use the Hardware Configuration tab to set the hardware configuration choices for your end user.



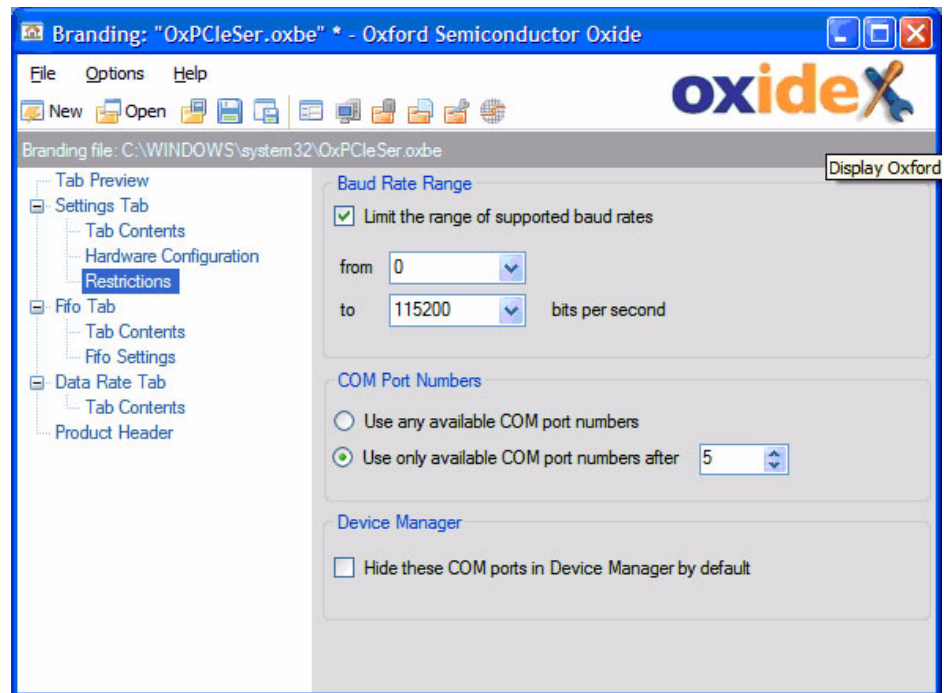
You can enable or prevent the end user selecting the hardware configuration. If you prevent them from choosing a hardware configuration, you can define which hardware configuration is used. If you select hardware configuration RS422/485, you can disable RS485.



Some configurations leave no user interface in the hardware configuration section. In this case, a warning appears, recommending you turn off the section entirely.

Restrictions

You use the Restrictions tab to set further driver installation restrictions.

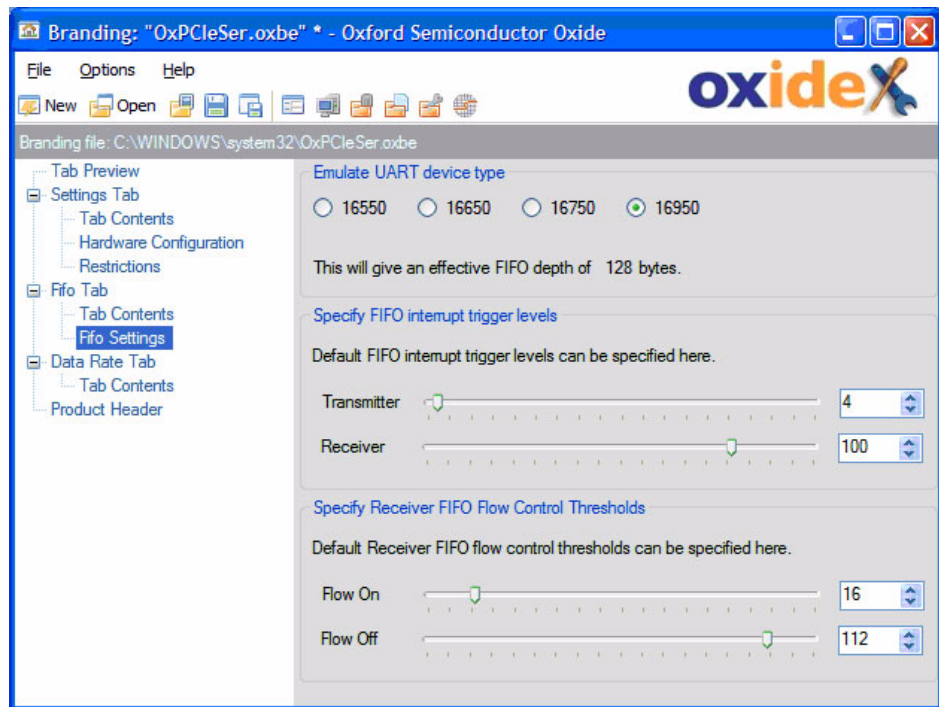


You can:

- Restrict the range of baud rates provided to the end user for this device
- Set the lowest COM port number allocated by Windows
- Hide COM ports in the Windows Device Manager. This can be useful for ports with devices permanently attached on a card, such as a modem. The end user can then see only the modem device and not the UART to which it is internally connected

FIFO Settings

You can restrict the use of the FIFOs, using the FIFO Settings tab.



You can restrict the effective FIFO depth available to the UART driver, so that the functionality of other UART devices, such as the 16550, can be emulated.

To set the default values for the FIFO interrupt trigger levels and Receiver FIFO flow control thresholds, in the left hand pane, click **FIFO Settings** under **FIFO Tab**. You can remove the ability to change these settings using the fields in **Tab Contents** under **FIFO Tab**. If you do remove the ability to change these settings, the values you enter are permanently used by the driver.

Branding File Completion

When you are happy with the appearance and capabilities of the branded UART driver, you need to include the changes in the master driver package. The simplest way to do this is to use **File - Save As**, then browse to the PLX Technology folder on your desktop. In this folder, browse to and select either the 32 bit or 64 bit branding file and overwrite it. You are prompted to overwrite the second branding file as well.

Now install the updated driver package to continue with the evaluation workflow.

Preparing the Final Driver Package


To create a deliverable driver package, you must create a clean driver package without any of the information used by Oxide to perform its operations. For more information on preparing the driver package, see [Creating the Driver Package](#) below and [Creating a Final Driver Package](#) on page 68.

You can also create a driver installer; for more information on this, see [Creating a Driver Installer](#) below and Chapter 7 [Installer Creation](#).

Creating the Driver Package

To support the Oxide INF editing features, PLX Technology .inf files contain comments describing the content that should be presented for editing. This information has no impact on the functionality or installation of the driver.

To tidy the .inf files and remove the comments, use the INF editor to open the

driver package in the PLX Technology folder on your desktop. Select  **Create Release Package**. This makes a copy of the entire driver package folder in your selected location and creates .inf files without the comments.



It is important to retain a master copy of the driver package that contains the Oxide information; otherwise you cannot make further changes using Oxide.

Signing the Driver Package

If you have set Oxide to automatically create catalog files and digitally sign the driver, the released driver package is signed when you save it (see [Setting up Digital Signing](#) on page 6).

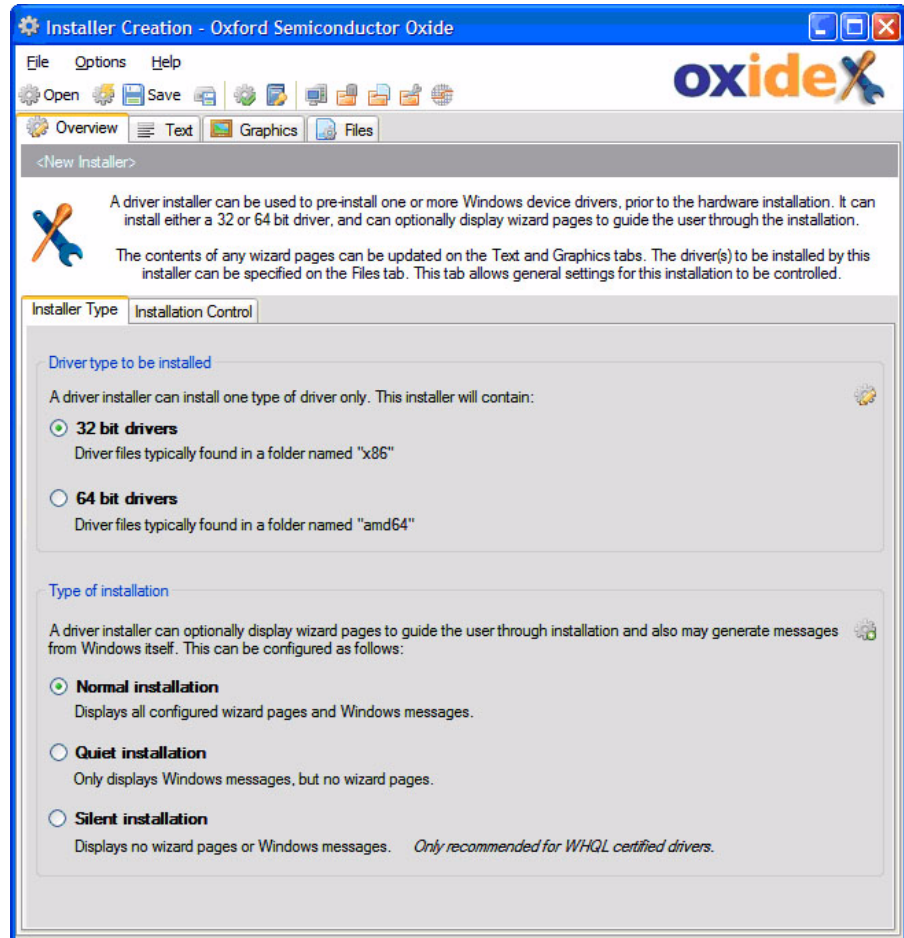
If you need Microsoft WHQL certification for your driver package, you must test and certify the package now. For more details about WHQL certification, see the Microsoft Winqual website. Microsoft sends you updated catalog files to replace those in your driver package.

Creating a Driver Installer

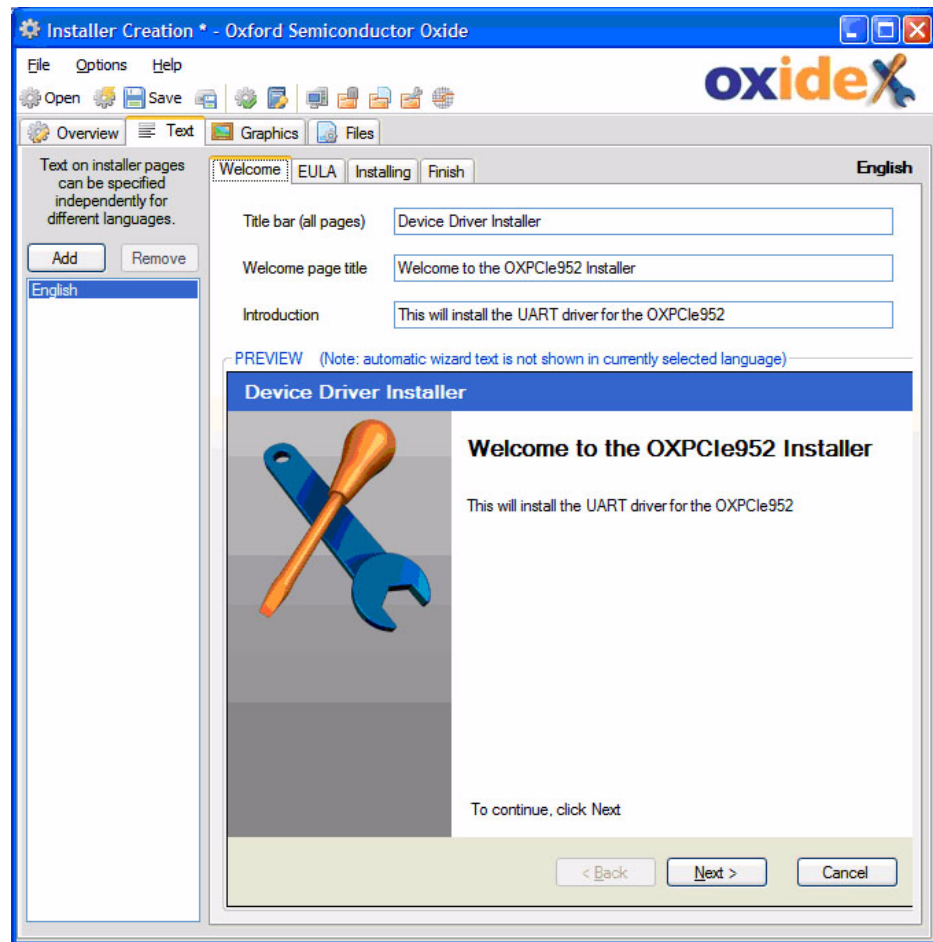
The final driver package you created can be installed after your customer has inserted the board and it has been detected by Windows. If you want your driver to be installed before your customer inserts your card, you can distribute your 32 or 64 bit driver as an installer. To create and configure your driver as an installer, use the Oxide Installer Creation utility.

To create an installer:

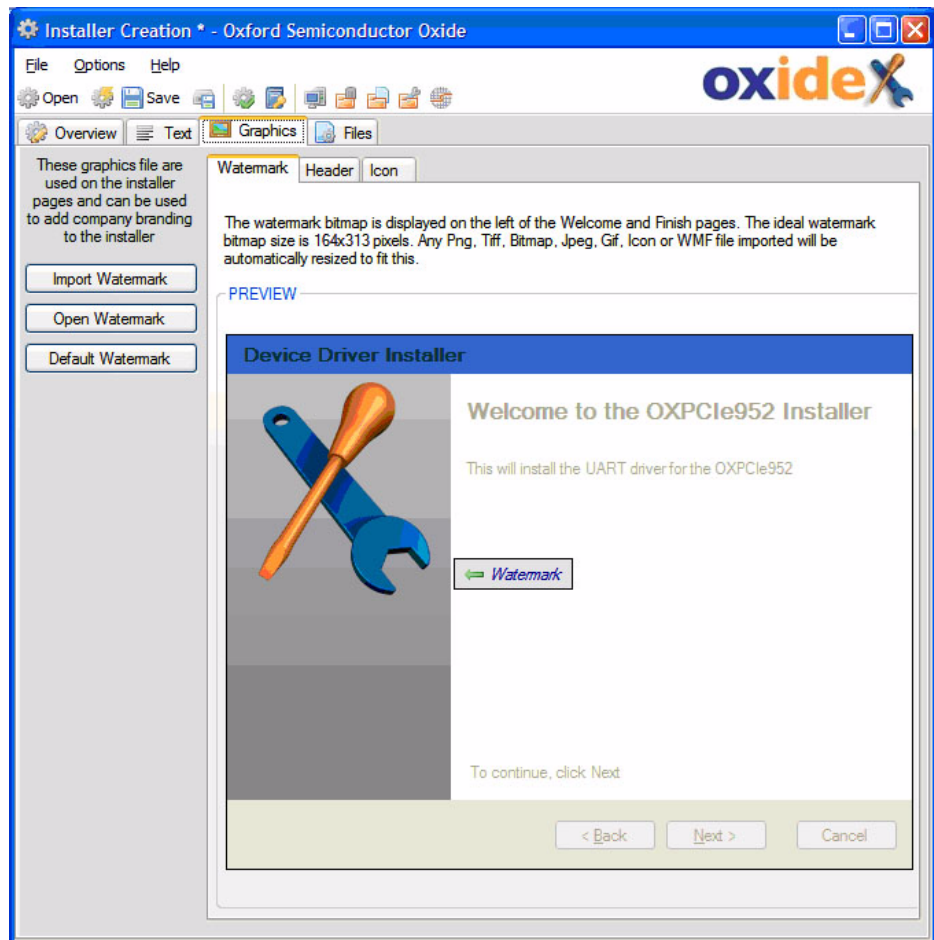
- 1 On the Evaluation workflow window, click **Installer Creation** to show the Installer Creation window. The Overview tab is shown, with the Installer Type secondary tab displayed.



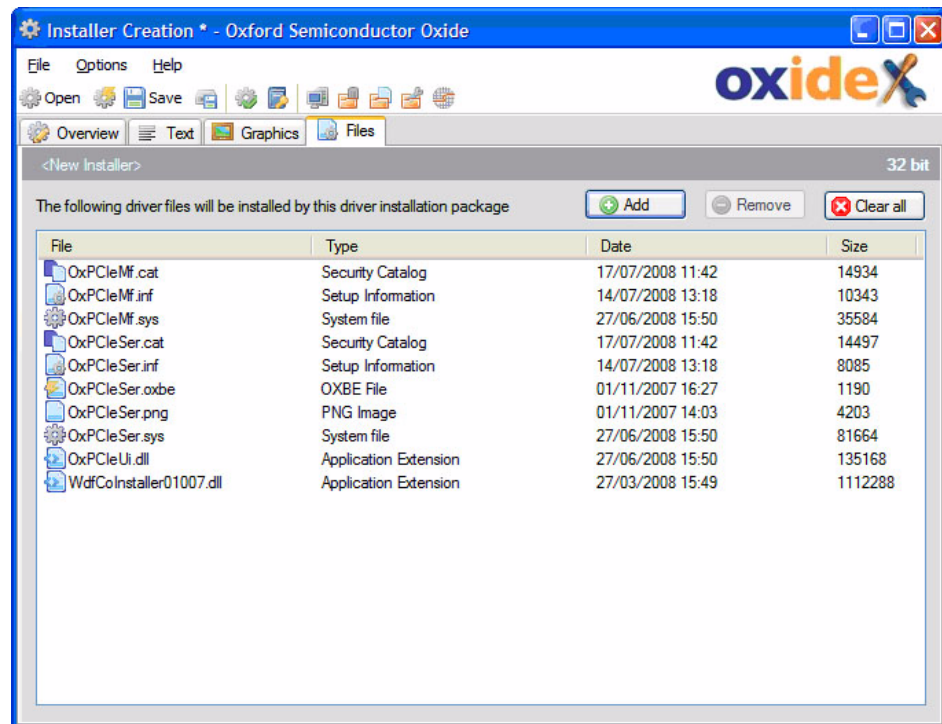
- 2 To set the bit size for the operating system your driver is to be installed on, under **Driver Type to be installed**, click the relevant radio button for your driver.
- 3 To set the level of user feedback required during installation, under **Type of Installation**, click the relevant radio button.
- 4 Advanced installation and uninstallation settings can also be controlled using the Installation Control tab. For more information, see [Chapter 7 Installer Creation](#).
- 5 To define the messages displayed on each wizard page displayed during driver installation, click the Text tab to display it.



- 6 You can set your installer to display text in different languages. To add a language, click **Add** and choose the language.
- 7 For each supported language, use the secondary tabs to specify the text and messages you want.
- 8 To specify the graphics used on your installer wizard (such as your company logo), click the Graphics tab to display it.



- 9 To import the graphics, use the relevant secondary tabs and follow the instructions on the screen.
- 10 To specify the driver files to be installed by the driver installer, click the Files tab to display it.



- 11 Click **Add** and browse to the folder containing the files for your driver package (typically in the **x86** folder for 32 bit drivers or **amd64** folder for 64 bit drivers). All contents of the folder are added.
- 12 To install more than one driver using the same installer package, click **Add** again and browse to another folder containing the driver files you want. Do not mix 32 and 64 bit drivers in the same installer.

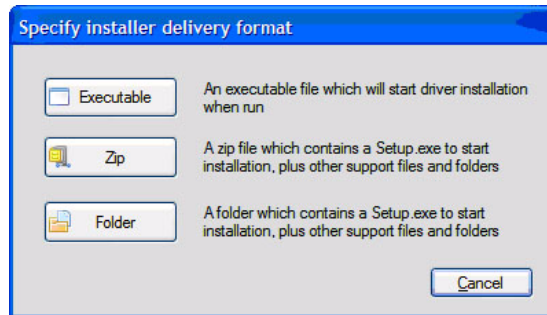
Testing and Saving the Installer

To test the driver installer on your system before you save it, go to the tool bar and click **Test Installer** or use **Options - Test Installer**.

Windows installs the driver and you see the user interface you set up (installer windows, messages, or no feedback if you chose **Silent Installation** when [Creating a Driver Installer](#)).

To save the driver installer:

- 1 Select **File - Save** or **File - Save As**. The **Specify installer delivery format** window is shown.



- 2 Click the button for the form you want to distribute the driver installer in, or click **Cancel** to cancel the creation.
- 3 When prompted, browse to the folder you want to save the installer in. The files are saved in the format you chose.

This page is intentionally blank



Design Base

The Oxide Design Base provides information relating to supported PLX Technology devices. It gives you easy access to technical documents such as data sheets, application notes and schematics, along with device drivers and other support collateral. The Design Base can automatically check for updates for newer versions from the PLX Technology website, and download updates when available.

If you need information to aid your evaluation or design-in using one of the supported PLX Technology devices, the Design Base should always be your first port of call.

The Design Base provides:

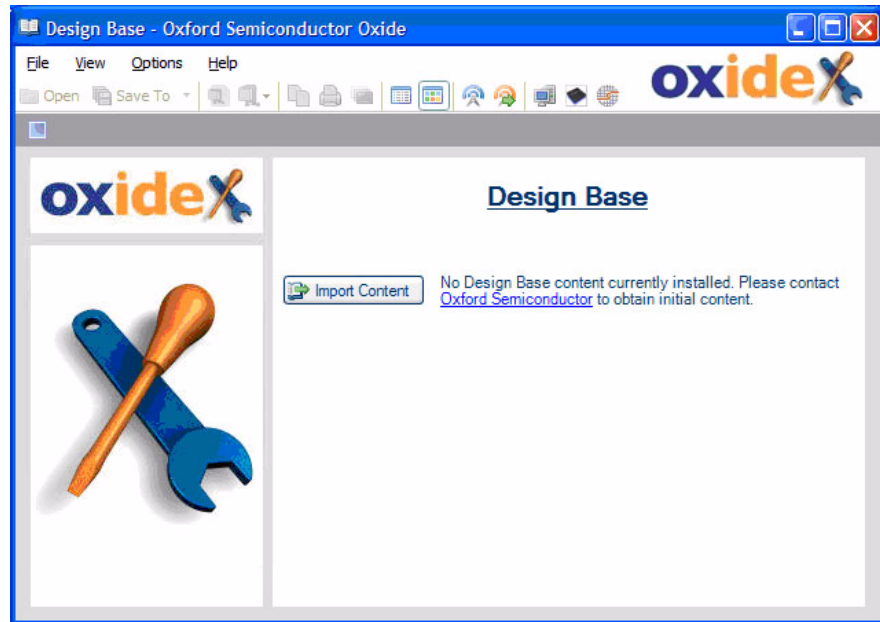
- Access to PLX Technology documentation and software, organized by product area and device
- Easy methods for extracting, opening or printing content on the Design Base
- Automatic updates to content from the PLX Technology website
- Access to previous versions of content for reference

Getting Started

To open the Design Base, go to the **Start** menu and click the Design Base shortcut, or go to the Launcher window and click **Design Base**. What you see the first time you open the Design base depends of how you installed Oxide: see [Initial View When Installed from the PLX Technology Website](#) and [Initial View When Installed Using an Oxide CD](#).

Initial View When Installed from the PLX Technology Website

If you installed the Design Base using an Oxide installer downloaded from the PLX Technology website, no Design Base content is initially available. In this case, a window similar to below is displayed.



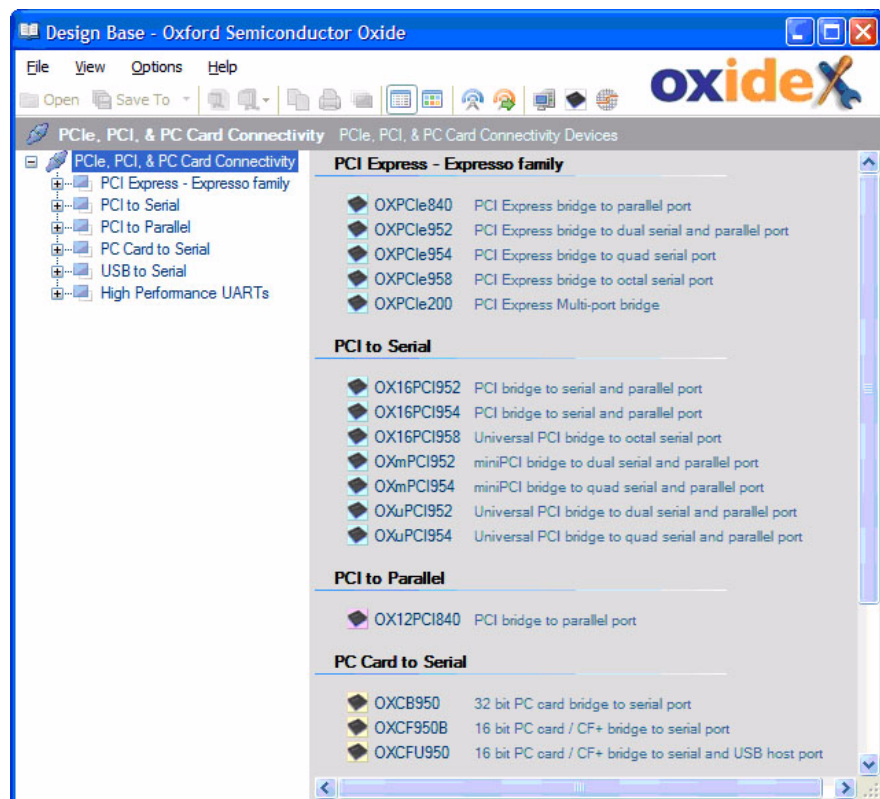
If you already have a Design Base content file to hand (a file with extension **.oxz**), click **Import Content** and select this file. If not, you must download the content from the PLX Technology [Oxide website](#).

Once you have downloaded the **.oxz** file to your local machine, click **Import Content** and select the file. At the end of the import, Oxide automatically checks the PLX Technology website for any additional updates to this content. For more information, see [Configuring and Downloading Updates](#) on page 44.

To load **.oxz** files at any other time, use **File - Import Content**.

Initial View When Installed Using an Oxide CD

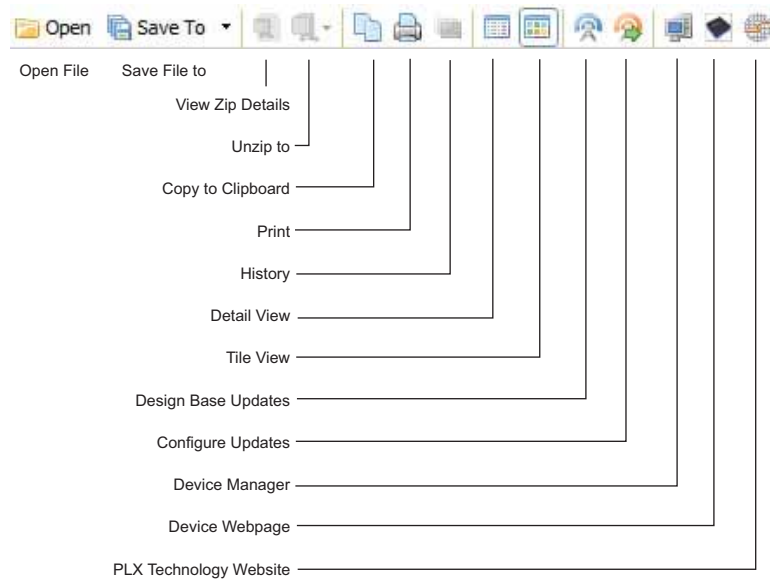
If you installed the Design Base from an Oxide CD, initial Design Base content is made available automatically. This is ready for immediate access, and the user interface is similar to below, showing the devices grouped by device type.



The main window has the same structure as many Oxide applications: a menu and tool bar at the top; a pane on the left where you select the device you want; and the main pane on the right showing the content for the selected device.

Design Base Tool Bar

The Design Base main window includes the tool bar shown below.



The tool bar buttons are:

- Open—opens a file you select in the right hand pane
- Save to—saves a copy of a selected file in a location you choose
- View Zip Details—displays details of zipped files; for more information, see [Handling Zip Files](#) on page 43
- Unzip to—extracts zipped files and saves them in a location you choose; for more information, see [Handling Zip Files](#) on page 43
- Copy to Clipboard—copies the file contents to the clipboard, for you to paste in a file you choose
- Print—prints the selected file
- History—displays the update history; for more information, see [Viewing the Update History](#) on page 46
- Detail View—displays Design Base contents with full details, as described in [Viewing Content](#) on page 41
- Tile View—displays Design Base contents in list format, as described in [Viewing Content](#) on page 41
- Design Base Updates—checks for updates to Design Base contents; for more information, see [Configuring and Downloading Updates](#) on page 44
- Configure Updates—configures how the Design Base contents are updated; for more information, see [Configuring and Downloading Updates](#) on page 44

- Facilities:
 - Device Manager—opens the Windows Device Manager
 - Device Webpage—opens the web page for the device on the PLX Technology website; for more information, see [Accessing the PLX Technology Website](#) on page 43
 - PLX Technology Website—opens the PLX Technology website

Design Base Menu Bar

The menu bar at the top of the Design Base main window allows you to control Design Base activity:

- **File**—to manage files in the Design Base; see [File Menu](#)
- **View**—to change the way the Design Base contents are displayed; see [View Menu](#) on page 39
- **Options**—to configure Design Base settings; see [Options Menu](#) on page 39

File Menu

The **File** menu options are:

- **Open**—to open a document or file
- **Save To**—to save a file to a location you choose
- **Copy to Clipboard**—to copy the contents of a file to the clipboard for pasting in a document you choose
- **Print**—to print a document or file
- **History**—to view the file history; for more details, see [Viewing the File History](#) on page 43
- **Import Content**—to import new Design Base content; for more details, see [Downloading Updates](#) on page 44

View Menu

Use the **View** menu options to modify the details displayed for files:

- **Detail View**—to display details with each file
- **Tile View**—to display files in a list

For more information, see [Viewing Content](#) on page 41.

Options Menu

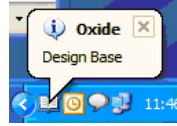
Use the **Options** menu to configure Design Base settings:

- **Run at Startup**—to automatically open the Design Base when you start up the PC; for more information, see [Configuration Options](#)
- **Minimise on Close**—to set whether the Design Base minimizes to the system tray or closes when you click Close; for more information, see [Configuration Options](#)
- **Design Base Updates**—to check for updates to Design Base contents
- **Device Manager**—to open the Windows Device Manager

Configuration Options

By default, the Design Base is set to run at startup, but minimized to an icon in the Windows System Tray. When you click **Close**, Oxide minimizes to the System Tray again, as shown in [Figure 3-1](#).

Figure 3-1 Design Base Icon in System Tray



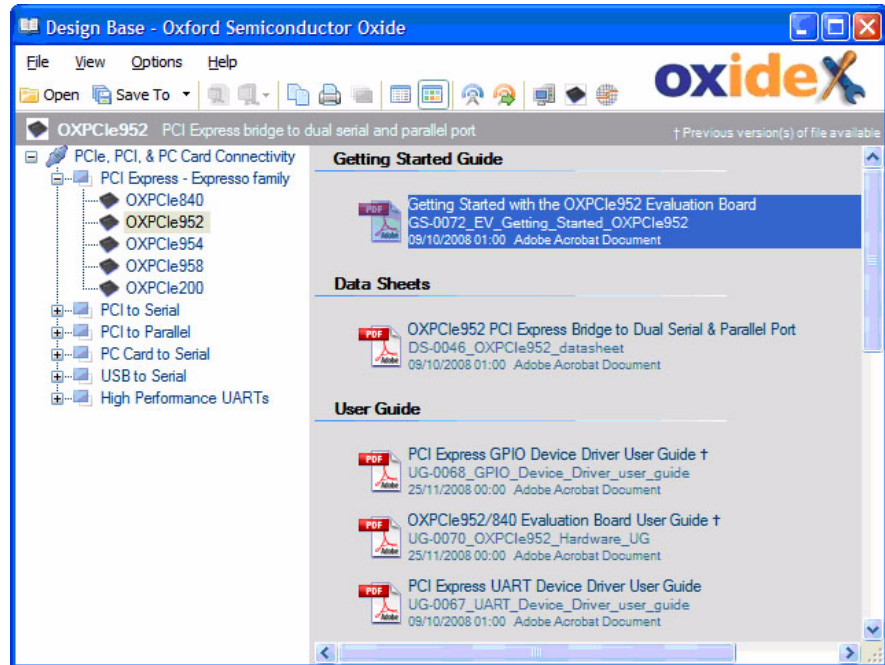
To close the Design Base completely, right-click the System Tray icon and select **Exit** from the menu, or use **File - Exit**.

We recommend that you keep the Design Base running, as it is always readily accessible, and it can periodically check for content updates, ensuring you always have the most up to date content available. To alter the default behavior, go to the **Options** menu and choose one of:

- **Run at Startup**—if unchecked, you must manually run the Design Base. No content updates occur unless it is running
- **Minimize on Close**—if unchecked, clicking **Close** closes the Design Base rather than minimizing it to the System Tray

Viewing Content

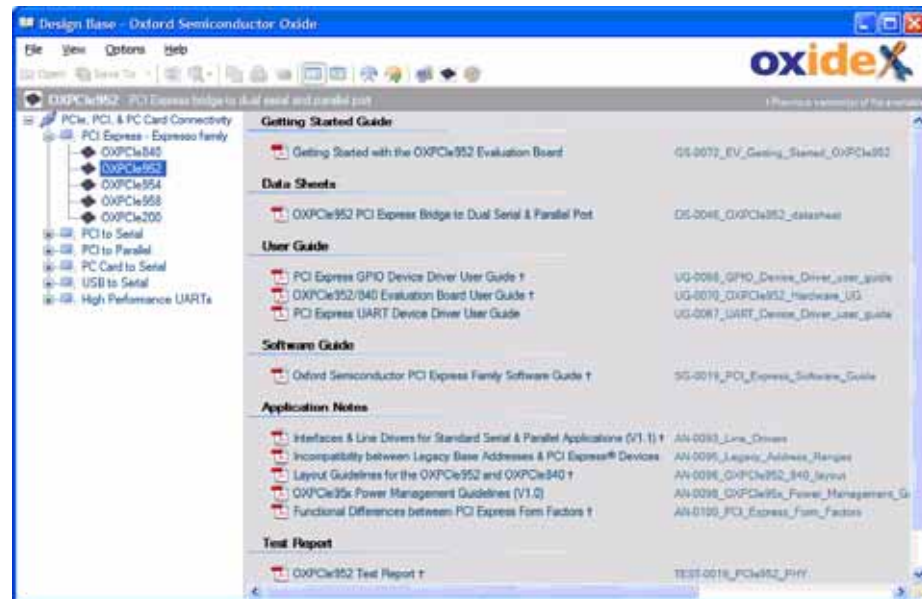
Supported devices are listed in the left hand pane of the Design Base window. To find the device you want, expand the product area of interest, locate the relevant device, and select it. The pane on the right shows the content available for this device.



The standard tile view shows each file organized by category of file. The details shown include:

- A full description of the file
- The file name
- File date and the file type
- An icon indicating the application needed to open the file in Windows
- If file version information is available, this is shown in parentheses after the file description
- If details are available on the changes made in the latest version of this file, these are displayed in a tooltip when the cursor is placed over the file

To view this information in a list view, use **View - Tile View**. The details are displayed.



When running Design Base under Windows 2000, the content is not grouped into file categories due to user interface limitations on this system. Instead the category is displayed as part of the data.

Accessing Content

The simplest way to access the content of a file is to double-click it. The file is opened in the default application for that file type on your PC (e.g. PDF files may be opened by Adobe Acrobat Reader).

You can also access multiple files by:

- Holding **Ctrl** and clicking each file you want, to select files at various points in the list
- Clicking a file, then holding **Shift** and clicking a file further down the list, to select all files in between

To work with the selected files, either click the tool bar button you want or right-click and choose the option you want from the menu. You can select:

- **Open**—to open each file in the default application for that file type on your PC
- **Save To**—to save a copy of the file to the selected folder. Shortcuts are provided to save the file directly to the desktop, My Documents or to a selected folder
- **Copy to Clipboard**—to place the file information on the Windows clipboard. To paste a copy of the files in a folder, use Windows Explorer to navigate the folder you want, and select **Paste**
- **Print**—to open and print the file. This option may not be supported by all applications for all file types
- **Drag and drop**—to drag the files from Design Base and drop a copy of the files into the folder you choose in Windows Explorer

Handling Zip Files

You can handle zip files in the same way as other files. Zip files often contain software, particularly device drivers. When you select a single zip file, the following additional options are available:

- **View Zip Details**—if the zip file contains a readme file giving more detailed information on the content of the file, this is displayed
- **Unzip To**—the zip file is unzipped to a selected folder. Shortcuts are provided to unzip directly to the Desktop and My Documents. In both cases, the contents are unzipped under an PLX Technology folder to aid co-location of multiple unzipped files. You can specify a different target folder. In all cases, once the unzipping is complete, the folder is automatically opened in Windows Explorer

Deleting Faulty Files

When you select a single file, an option is available to delete the file. This is for advanced use only. The physical file is deleted, but a reference to the file is kept in the Design Base. A new version is downloaded the next time an update check occurs. This option is provided in case you believe that your local file is corrupted, say due to a faulty download. This is a simple way forcing a new version to be downloaded.

Viewing the File History

When a new version of a file is stored in the Design Base, the previous version is not deleted; all older versions of the file are retained for reference. Content in the Design Base that has old versions is highlighted by the † marker next to the file description. You can do the following:

- To view the history of files with previous versions, select a file with the † marker; the **History** option becomes available when you right-click and on the tool bar.
To display a list of the previous versions of the file, select **History**
- To display a tooltip containing full information on the file version, hold the cursor over a file in the list
- To open this version of the file, double-click a file in the list
- To display a context menu with file options, right-click the file. The options are:
 - **Open**—opens the file in the default application for that file type on your PC
 - **Save To**—saves a copy of that version of the file to the desktop, My Documents or a selected folder
 - **Delete**—removes the old version of the file from the Design Base. Once deleted, this version of the file is lost from the Design Base and the operation cannot be undone. You cannot re-download an old version of the file

Accessing the PLX Technology Website

You can use options on the menu and tool bar for accessing information on the PLX Technology website using your default browser:

- To go to the home page, select **PLX Technology Website**
- To go to the web page for the selected device or product area, select **Device Web Page**. If there is no specific web page available, this takes you to the PLX Technology home page

Configuring and Downloading Updates

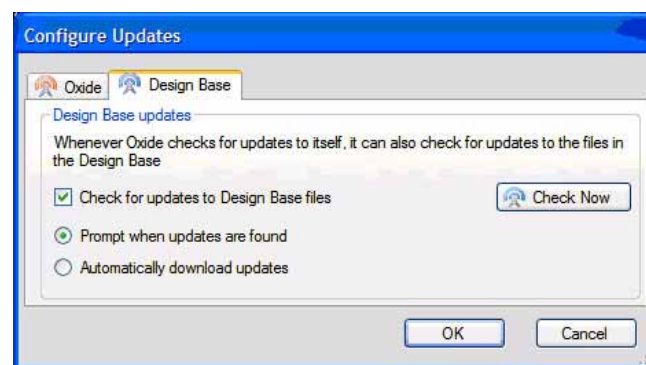
You can keep Design Base content up to date by downloading the latest information from the PLX Technology website. You can:

- Specify whether updates are automatically checked for and how often
- Choose to be notified of available updates (which you must manually download), or set updates to download automatically

For information on checking for and downloading Oxide updates (including design base updates), see [Configuring Oxide Updates](#) on page 5.

Configuring Design Base Updates

To configure Design Base updates, on the Design Base window tool bar, click **Configure Updates**. The Configure Updates window is displayed, with an extra tab.



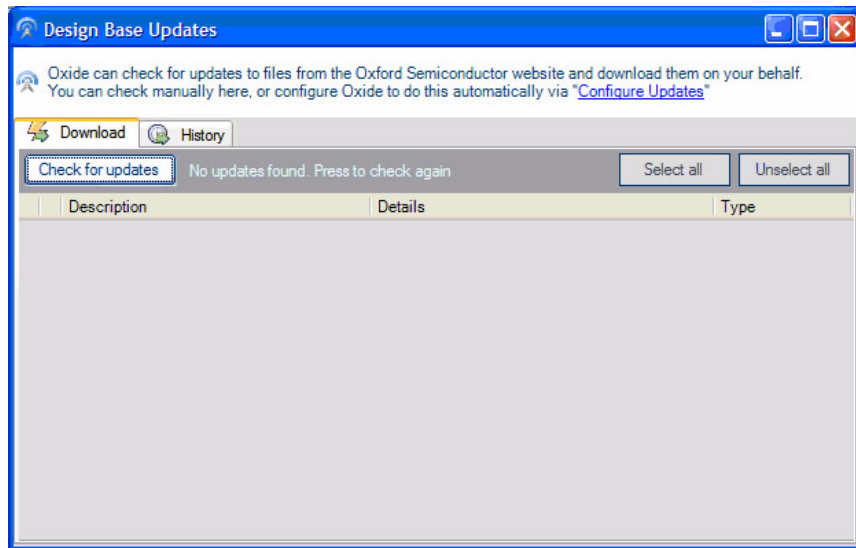
You can:

- Check for Design Base updates automatically, by checking **Check for updates to Design Base files**. Oxide checks for updates to itself and to the Design Base content
- For automatic checks, you can choose to be notified when updates are available for you to manually download, by selecting **Prompt when updates are found**
- For automatic checks, you can set updates to automatically download, by selecting **Automatically download updates**
- Disable the automatic check and manually check for updates, by unchecking **Check for updates to Design Base files**. To manually download updates, see [Downloading Updates](#)
- Check for available updates, by clicking **Check Now**. If there are no updates, a message is displayed indicating this. If there are updates, the Design Base Updates window is displayed; for more information, see [Downloading Updates](#)

Downloading Updates

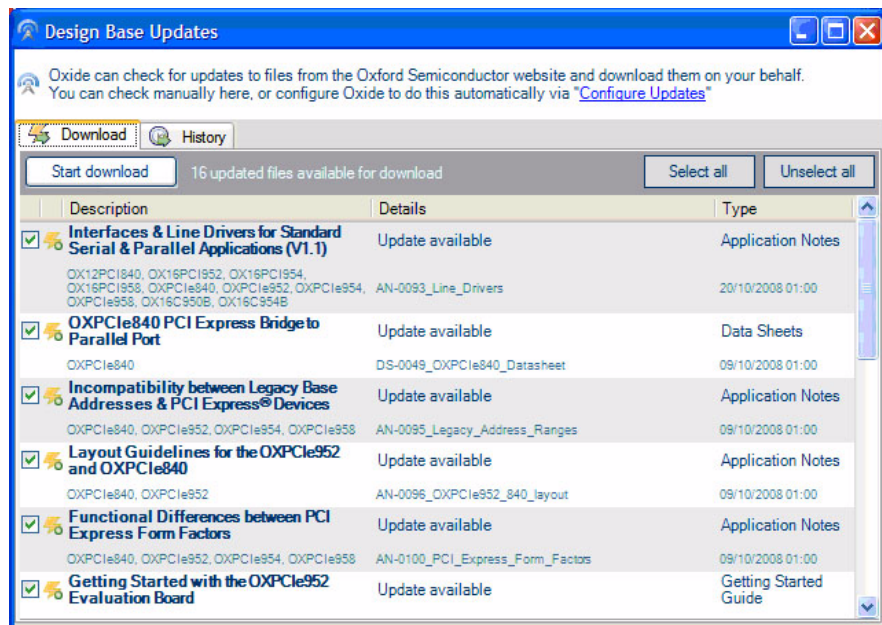
You can download updates using the Design Base Updates window, accessible from the **Options** menu, the tool bar, by pressing **F1** or when you click **Check Now** on the Configure Updates window.

When no updates are available, the window is similar to below.



To manually check for updates, click **Check for updates**.

If there are updates ready for download, the window displays relevant details.



All files are checked by default.

If you configured Design Base to automatically download updated files, the download starts automatically.

If you have not configured Design Base to automatically download updates, you can:

- Download all updates by clicking **Start download**
- Prevent downloading updates you do not want by unchecking the check box next to the unwanted update

- Use the **Select all** and **Unselect all** buttons to check and uncheck all the files with one click



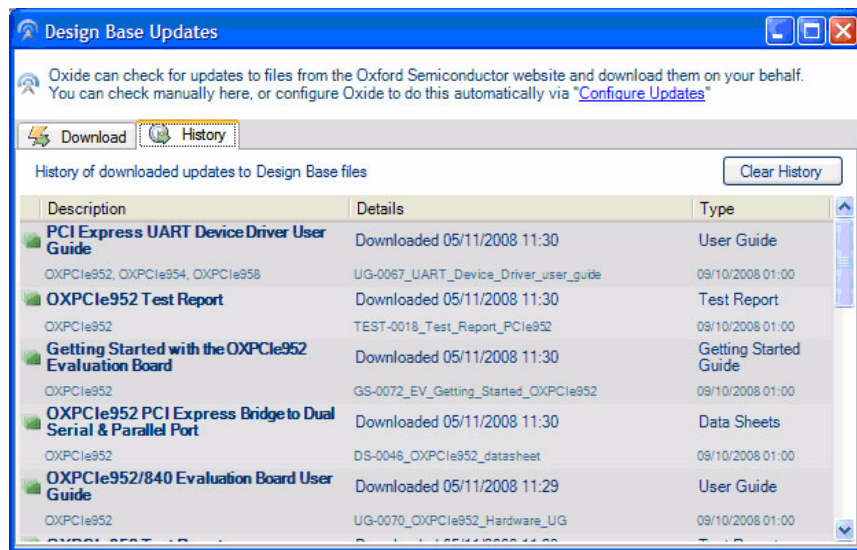
If you uncheck a file and do not download it, the update is listed again the next time you perform an update check.

The progress status of the download is displayed in the **Details** column.

To stop downloading, click **Cancel download**. Any partially downloaded file is deleted, and the original file retained in the Design Base.

Viewing the Update History

You can view details of downloaded updates by selecting the **History** tab in the **Design Base Updates** window.





When you configure Design Base to automatically check and download updates, this is an easy way to see the updates that have been downloaded. The most recent updates are shown at the top of the list.

You can clear the history information by clicking **Clear History**. Note that there is no way to recover this information once it is cleared. This does not affect the content of the Design Base in any way, including any old versions of files that have been retained.

EEPROM Editing

Using the PLX Technology EEPROM editing utility, you can configure devices by programming the EEPROM of a UART or bridge product. The EEPROM overrides the power-up settings of the device; for example, the vendor ID, device ID, local bus or power management settings or, for PC card devices, the card bus information structure (CIS).

The EEPROM editor is an excellent prototyping tool, as it speeds up the cycle of device configuration, EEPROM programming and testing. You can create new EEPROM images quickly and easily, based on either an image file or a golden image file; or you can modify and verify existing settings. You can save the resulting image file, write it to an EEPROM and test it, or write it to a succession of devices.

-  The EEPROM editor can handle image files in a variety of formats, including binary, hexadecimal and Motorola S-records. Golden image files are always in binary format.
-  Formats such as Motorola S-records and Intel hex can be used to program the EEPROM on a gang programmer.

To create the optimum interface settings for the device during EEPROM configuration, we recommend that you refer to the data sheet for your device.

You can also use OxShell (the PLX Technology command shell EEPROM programming tool) to write images created using Oxide and to create more advanced content. For more information on OxShell, see Chapter 8 [OxShell](#).

Programming EEPROMs with the EEPROM Editor

You can use the EEPROM editor for:

- Creating an EEPROM image file for use by the EEPROM editor or OxShell
- Creating an EEPROM image file for use by third party EEPROM programmers
- Programming an EEPROM from scratch and saving the image file for reuse
- Programming an EEPROM from scratch, using (and modifying) an existing EEPROM image file
- Modifying the EEPROM contents for a device
- Erasing the EEPROM contents for a device
- Reading the current EEPROM image from a device and saving it to an image file for reuse
- Verifying the EEPROM image in a device against a loaded image

Starting the EEPROM Editor

To start the editor, either select **EEPROM Editing** from the Windows start menu or click **EEPROM Editing** on the Launcher window.

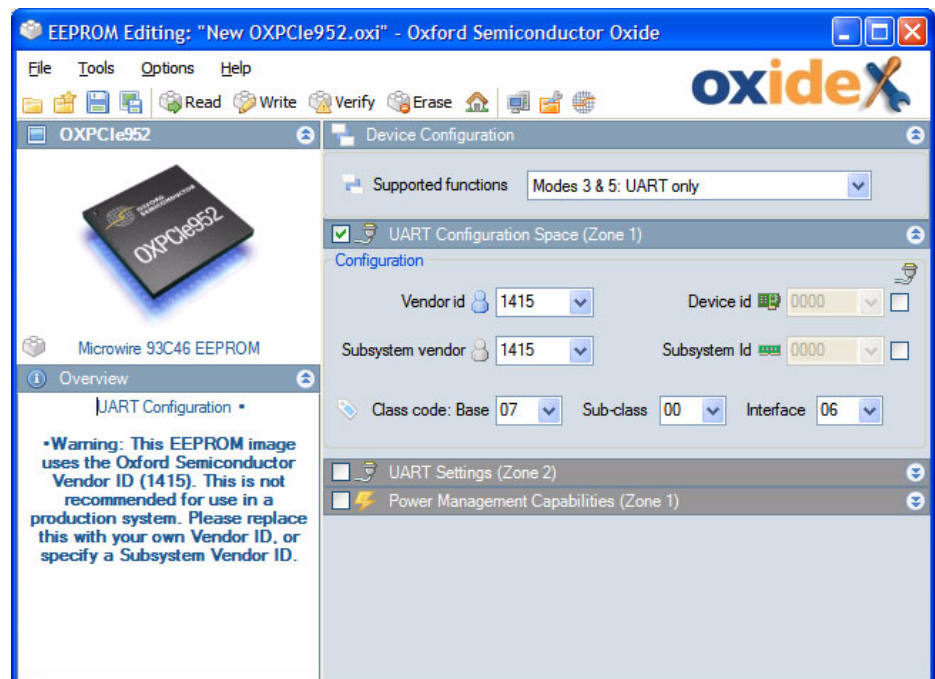
It is not mandatory for a device to be present when the editor starts up. If you are running the utility for the first time, it asks you to select a device.



This shows a typical display, which appears when a PLX Technology device is attached to the bus. If there is no device, the **EEPROM Type** drop down list does not appear.

The EEPROM Editing main window is shown. If you have already used the EEPROM editor, the window is populated with the details of the last golden image you used.

The screen usually shows the EEPROM type below the graphic if the device is attached to the host.



Now you can either work with the golden image settings or load the EEPROM contents from the connected device or from a file and work with them.

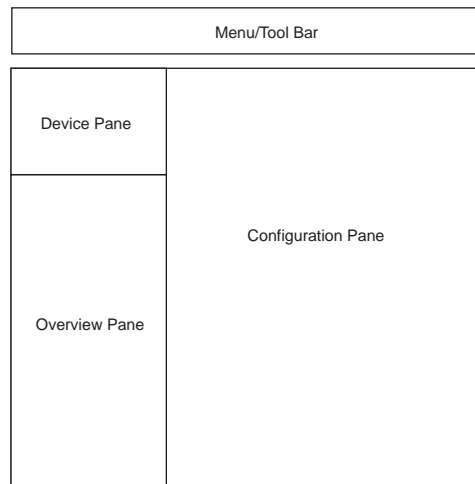
Golden Image Files

PLX Technology supplies at least one golden image file for each supported device. Golden image files contain typical configuration settings for the device in binary format. To select and load a golden image, use **File - Open Golden Image**. You cannot overwrite a golden image file using the EEPROM editor.

Using the EEPROM Editor

You use the EEPROM Editing main window to modify EEPROM content. Window elements and file options that do not apply to the current device are grayed out.

The EEPROM Editing main window is divided into panes.

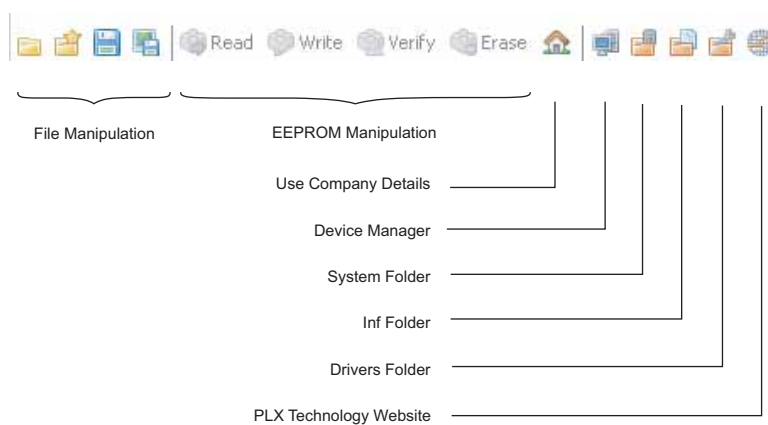


The panes display information as follows:

- Menu and tool bar—shortcuts to common activities, as explained in [EEPROM Editor Tool Bar](#) on page 51
- Device pane—a graphic of the device currently being configured. If the device is attached to the bus, details of the EEPROM type are also displayed. To change the device, click the device graphic; this opens the **Select Device Type** window shown in [Starting the EEPROM Editor](#) on page 48
- Overview pane—a summary of the device information in the configuration pane. This pane also indicates whether a device is attached to the bus. The contents of the overview pane change when you select different options in the configuration pane
- Configuration pane—typically showing the following information for the device:
 - Device configuration settings, which are explained in more detail in [Device Configuration Settings](#); these are only required for some devices
 - EEPROM zones, depending on the device and mode, which are explained more fully in [Zone Settings](#) on page 55. The zones contain information such as PCI configuration space settings, CIS settings and function-specific controls

EEPROM Editor Tool Bar

The EEPROM Editing main window includes the tool bar shown below.





The tool bar buttons are:

- Standard file handling—open, close, save and save as options
- EEPROM manipulation—these options are grayed out when no device is attached to the bus
- Facilities:
 - Use Company Details—populates the fields with company details in the current EEPROM image, or opens the Company Details window if you have not entered company details. For more information, see [Company Details](#) on page 6
 - Device Manager— opens the Windows Device Manager
 - System folder—opens the Windows system32 folder
 - Inf folder—opens the Windows inf folder
 - Driver Folder—opens the Windows Drivers folder
 - PLX Technology Website—opens the PLX Technology website


Screen Icons

Some icons on the EEPROM editor window are active and can be used as follows:

-  —to collapse and expand a section
- Clicking anywhere in the title bar of a pane also collapses and expands a section
-  —opens either the PCMCIA or PCI website, for checking IDs



The default EEPROM image uses the PLX Technology manufacturer ID and vendor ID, which is not recommended for use in a production system. You must replace them with your own ID information. The PCMCIA and PCI-SIG websites contain information on IDs and how to register them.

-  —opens the PCI-SIG website to access the PCI specification for class code, subcode and interface values

EEPROM Editor Menu Bar

The menu bar at the top of the EEPROM Editing main window allows you to control EEPROM programming activity:

- **File**—to manage a library of EEPROM configuration files; see [File Menu](#)
- **Tools**—to manipulate the EEPROM on a connected device; see [Tools Menu](#) on page 53
- **Options**—to find relevant information for configuration settings; see [Options Menu](#) on page 53

File Menu

The contents of the EEPROM editor window vary depending on whether a device is attached to the bus and, if so, what type it is. Even if no device is attached to the bus, you can load configuration information from a file for a supported device and work with it using the **File** menu options:

- **Open Image**—to open a previously created image file
- **Open Golden Image**—to open a golden image to program a device; this opens the **GoldenImages** folder for the current device
- **Save**—to save an image file (but not a golden image file)
- **Save Image As**—to create a new image file from a modified image file or golden image file

Tools Menu

Use the **Tools** menu options to modify the device EEPROM settings:

- **Read EEPROM**—to read the EEPROM contents from a device
- **Write EEPROM**—to write an EEPROM image back to the device
- **Verify EEPROM**—to verify the EEPROM contents against an image file or a golden image file
- **Erase EEPROM**—to erase the EEPROM contents of a device

Options Menu

Use the **Options** menu to select devices to use for EEPROM configuration:

- **Select Device**—to change the device by selecting from a drop down list
- **Select EEPROM Type**—to change EEPROM type for a device; see [Selecting an EEPROM](#) on page 54



This option is grayed out when you are working without the corresponding device on the bus, or for devices with internal EEPROMs.

- **Device Manager**—to open the Windows Device Manager
- **System Folder**—to open the Windows System 32 folder
- **Inf Folder**—to open the Windows inf folder
- **Drivers Folder**—to open the Windows Drivers folder

EEPROM Manipulation

This section describes how to edit EEPROM contents.

Selecting a Device

To select a device, either use **Options - Select Device** or click the device image in the device pane to open the Select Device Type window (see [Starting the EEPROM Editor](#) on page 48). You also need to specify the EEPROM type for the device, if it has an external EEPROM.

If you select a device with an internal EEPROM, such as the OXCFU950, the EEPROM drop down list is grayed out.

If no device is attached to the host, menu options for manipulating the EEPROM, such as **Read EEPROM**, are disabled.

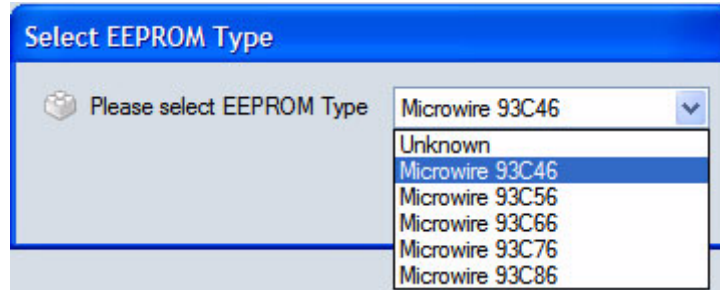


To access the EEPROM, you must ensure the device is present and the relevant PLX Technology device driver is installed.

If you have unsaved changes when you change the device type, the EEPROM editor prompts you to save the current image to a file before proceeding.

Selecting an EEPROM

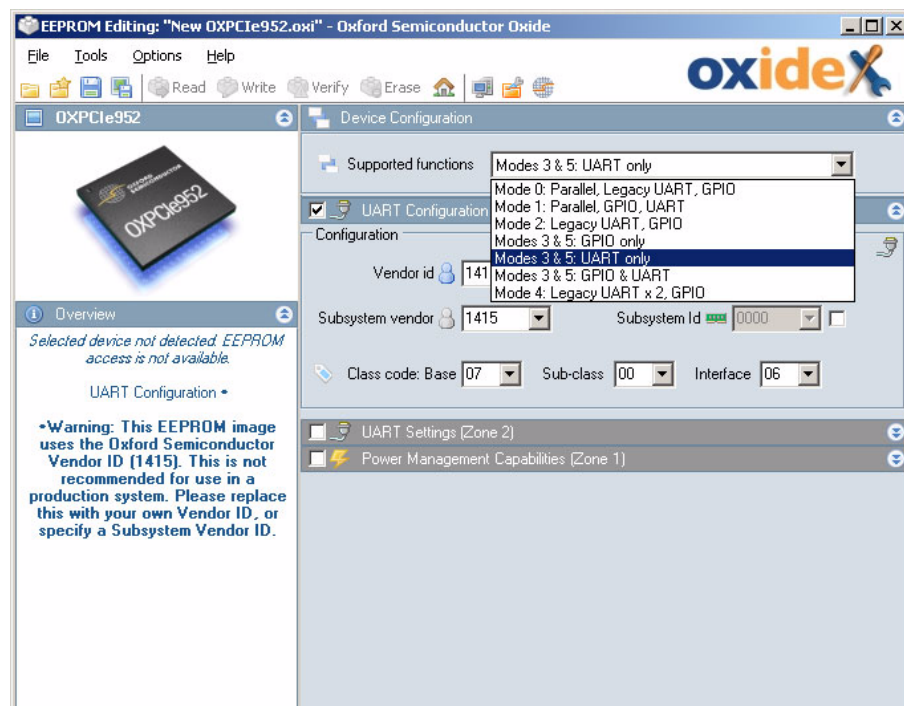
You can specify a different EEPROM type for a device by using **File - Select EEPROM Type** to display the **Select EEPROM Type** window.



Device Configuration Settings

Some PLX Technology devices can use several combinations of interfaces. In this case, the available functions are listed in a drop down list in the configuration pane. For example, the screen below shows the options available for the PLX Technology OXPC1e952, which is Mode 3 or Mode 5—modes for which only native UARTs are required. Options for other PLX Technology devices are likely to be different.

The configuration options available for a device are fully documented in the device data sheet.

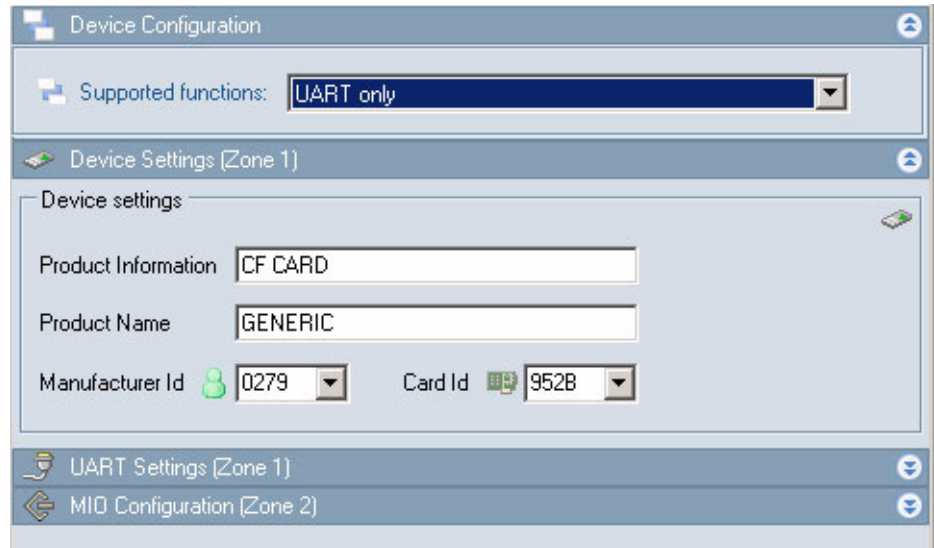


For some PLX Technology devices, there is only one possible device configuration, so the Supported Functions drop down list is grayed out. For these devices only the **Zone Settings** are displayed.

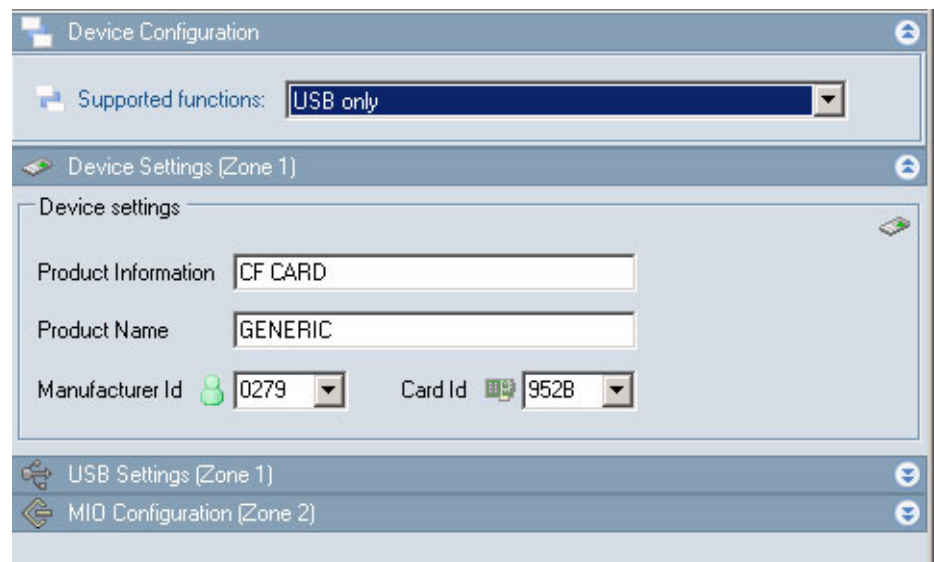
Zone Settings

Selecting a device, or a particular interface combination for a device, opens zone setting sections appropriate to the selection.

For example, selecting the interface combination **UART and USB** for the OXCFU950 opens UART zone settings, USB zone settings and MIO zone settings; however, selecting **UART only** eliminates the USB zone settings.

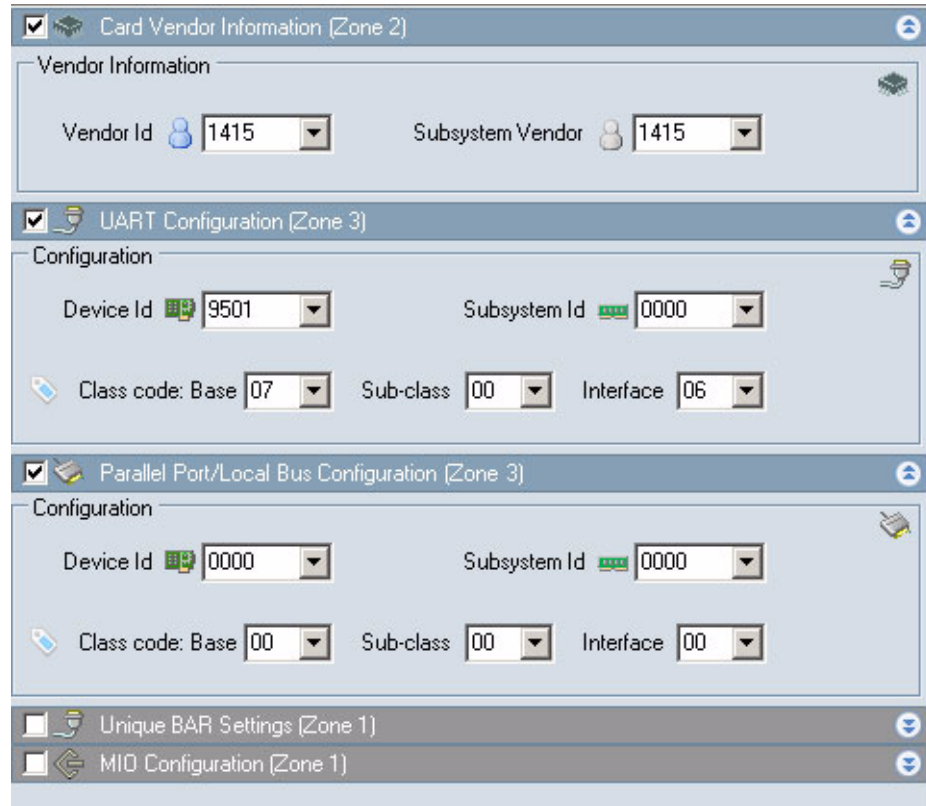


Selecting **USB only** eliminates the UART zone settings.



Where the zone settings are too large for the configuration pane, a scroll bar appears to give you full access.

Many zone sections display a check box on the left of the section title bar; to enable fields for that zone setting, you must check the box.



Editing EEPROM Settings

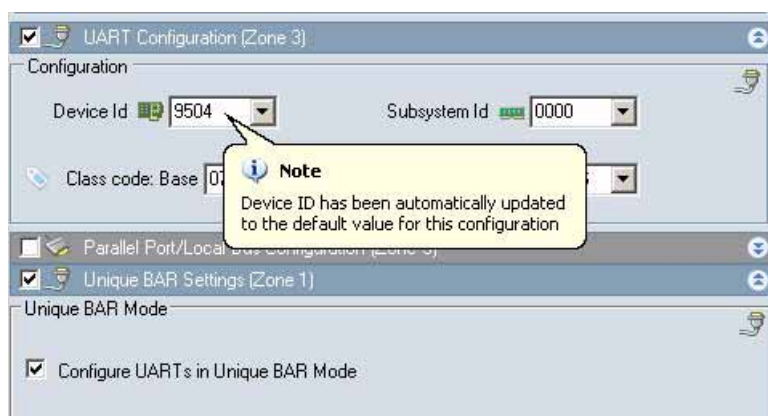
Selecting a device type and EEPROM causes the corresponding golden image to be loaded and determines the format of the configuration information to be written to the EEPROM.

You can modify the default golden image settings and any default parameter settings in the configuration pane sections.



There are several golden images for some devices, such as the OXuPCI954, which provide alternatives to the typical configuration. To see the golden images for the current device, click **Open Golden Image**.

In some cases, the EEPROM editor infers information from associated parameter settings and updates other configuration fields; for example, the screen below shows what happens when you check **Configure UARTs in Unique BAR Mode**.



Saving an Image File

When you have finished, you can save an image file:

- In binary format to a new image file (you cannot overwrite a golden image)
- In hexadecimal format
- In Motorola S-record or Intel hex record format for use in a gang programming utility

Files created by the EEPROM editor can be used by OxShell for bulk EEPROM programming in command line mode. For more information on OxShell, see Chapter 8 [OxShell](#).

Programming an EEPROM

To program an EEPROM, you load the image you want into the EEPROM editor. The image can be a modified file originally loaded from the EEPROM of a device on the bus, an image file or a golden image file.

Before writing the image to the target, the EEPROM editor checks that the correct device is present on the bus.

To program the EEPROM, click the **Write** tool bar button, or use **Tools - Write EEPROM**.

This page is intentionally blank

INF Editing

The Oxide INF editing utility is used to prepare a driver package for release to end users. A device driver package is a folder (and any subfolders) containing one or more .inf files, along with all the other files that are installed with a device driver.

.inf files are used by Windows to install a device driver for one or more devices. They contain information such as the vendor and device IDs the driver applies to, and the name displayed in Windows Device Manager for the device. The INF editor enables you to update selected contents of .inf files within a driver package without specialist knowledge of the .inf file structure.

The same information, such as the manufacturer name, is often used across multiple .inf files in the same driver package. The INF editor enables you to enter the information only once and automatically uses it in all appropriate .inf files.

You can edit .inf files at any stage in the driver package development, but you must only create the final driver package once you have finished all other development tasks. This includes initial setup (see [Initial Setup](#) on page 5), EEPROM editing (see Chapter 4 [EEPROM Editing](#)), .inf file editing (described in this chapter) and branding (see Chapter 6 [Driver Branding](#)).

You can create an installer using your driver package; for more information, see Chapter 7 [Installer Creation](#).

Managing .inf Files

You can use the INF editor to:

- Modify .inf files in an existing driver package
- Create a new driver package from an existing package
- Rename files in the driver package
- Digitally sign a driver package

.inf File Additional Information

.inf files contain additional information, in the form of comments, that describes the content Oxide presents for editing when you work on a driver package. This information has no impact on the functionality or installation of the driver. To remove this additional information from the .inf files to be supplied to an end user, see [Creating a Final Driver Package](#) on page 68.

Starting the INF Editor

To start the INF editor, either select **INF Editing** from the Windows Start menu or click **INF Editing** on the Launcher window. It is not mandatory for a device to be present to use the INF editor.

When run for the first time, the INF editor displays the Signing Setup window to allow you to specify whether catalog files must be generated and digitally signed when you save a driver package. For more information, see [Digitally Signing a Driver Package](#) on page 67. If you do not want to configure this immediately, click **Cancel**. You can configure this later by using **Options - Driver Signing Setup** or the equivalent tool bar button.

The initial INF Editing window is shown.



The options are:

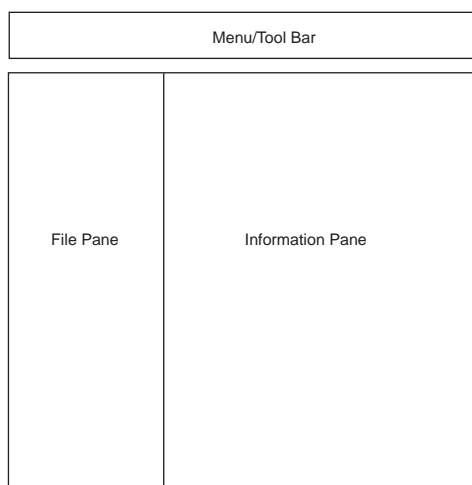
- **Open Last Package**—not available when you first run the utility. Click to display .inf information for the last package that was edited
- **Open Package**—to choose .inf files for the driver package you want. A Browse for Folder window opens. Navigate to the folder you want: the information in the .inf files in the selected folder is displayed for review and modification
- **Create New Package**—to create a new driver package based on an existing one. A Browse for Folder window opens. Navigate to the folder you want. The INF editor prompts you to choose a folder for the new package; you can create a new folder for the package if required

All the above options open the INF Editing window.



Using the INF Editor

The INF Editing window is divided into panes.

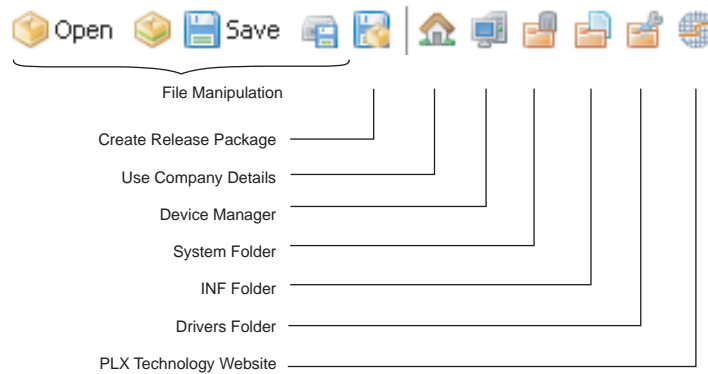


The panes display information as follows:

- Tool bar—short cuts to common activities, described in [INF Editor Tool Bar](#)
- File pane—.inf files in the package folder and an entry Shared information which signifies the common entries. For details, see [File Pane Contents](#)
- Information pane—typically showing the following for the package:
 - Shared information; which is described in [Shared Information](#)
 - .inf file entries, each with its own section, which are described in [.inf File Settings](#) on page 65

INF Editor Tool Bar

The INF Editing window includes the tool bar shown below.



The tool bar buttons are:

- Standard file handling—open package, create new package, save and save as options
- Driver release package creation—see [Creating a Final Driver Package](#) on page 68 for details
- Facilities:
 - Use Company Details—populates the fields with company details in the current EEPROM image, or opens the Company Details window if you have not entered company details. For more information, see [Company Details](#) on page 6
 - Device Manager— opens the Windows Device Manager
 - System folder—opens the Windows system32 folder
 - Inf folder—opens the Windows inf folder
 - Driver Folder—opens the Windows Drivers folder
 - PLX Technology Website—opens the PLX Technology website

INF Editor Menu Bar

The menu bar at the top of the INF Editing window allows you to manage INF handling:

- **File**—to manage the .inf files and create a driver release package; see [File Menu](#)
- **Options**—to access useful folders

File Menu

Even if no device is attached to the bus, you can open a driver package. The **File** menu options are:

- **Open**—to open an existing driver package and modify it
- **New**—to create a new package based on an existing one and modify it as required. This option makes a copy of the complete contents of the selected driver folder

File Pane Contents

File pane contents are initially displayed collapsed, but clicking the + expands the display.



By clicking the top of the hierarchy (UART in the image above), the package settings are displayed in sections in the information pane. By clicking the appropriate entry, you can display the settings for a specific .inf file or display only the shared information.



It is not usually necessary to expand the file pane contents. This is only for advanced use if it is important to know which content is relevant to which .inf file. For normal operation, leave the file pane contents collapsed with the top item selected.


Information Pane Contents

The INF editor information pane shows information relevant to the item selected in the file pane. When you select the top level item in the file pane, the information pane shows all the information that can be edited from all .inf files in the package.

When the file pane is expanded, selecting individual items from the hierarchy restricts what is shown. The information displayed can be:


- Shared information, which may be used by more than one .inf file in the package
- .inf file settings, which are only used in a single .inf file in the package

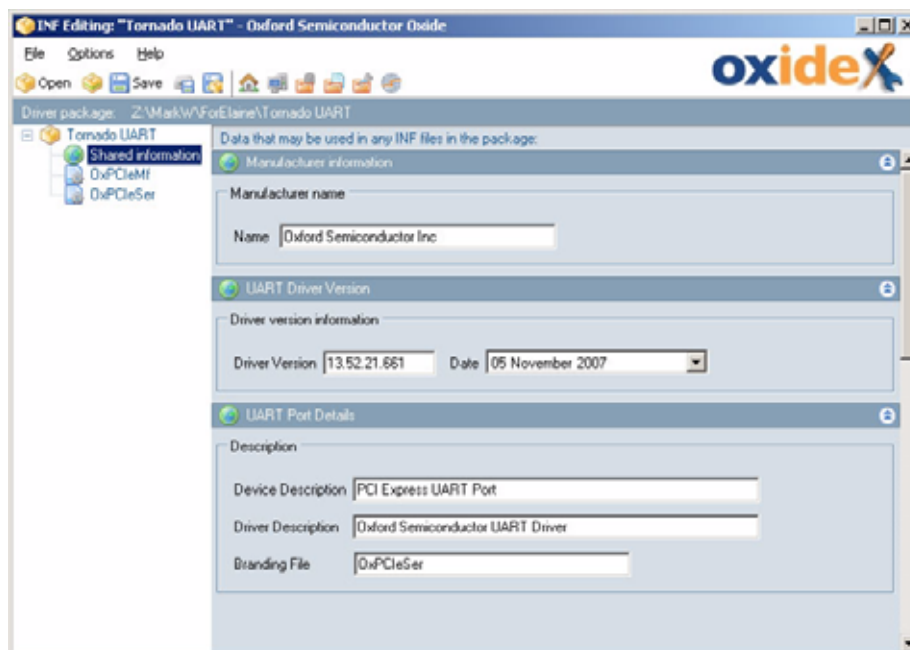
By default, some sections of information are collapsed and show only the title bar, while others are expanded. To toggle the information on display, do one of the following:

- Click  in the section title bar
- Click anywhere in the section title bar

Shared Information

Although the contents of an .inf file vary according to the device and how it is used, an .inf file can also contain shared information that may be used in any .inf files in the package, such as driver version number or manufacturer. For shared

information, the section title bar shows the icon . The INF editor displays shared information in its own section.



You can update the shared information as required, and the new settings are written back to the .inf files when you save the package. This simplifies the task of maintaining consistent information across multiple .inf files in the same driver package.

.inf File Settings

For every .inf file in the device package, the INF editor displays a section showing the settings for that .inf file.



All driver packages contain at least one section. As well as descriptions of the device and driver installed, there is a list of devices supported by the driver package.






If the window is too small to display all the details, a scroll bar appears to allow access to all the information.

For PCI and PCI Express devices, the list contains all vendor and device IDs for the devices to be supported by the driver package. For PCMCIA devices there is a section for manufacturer and card IDs.

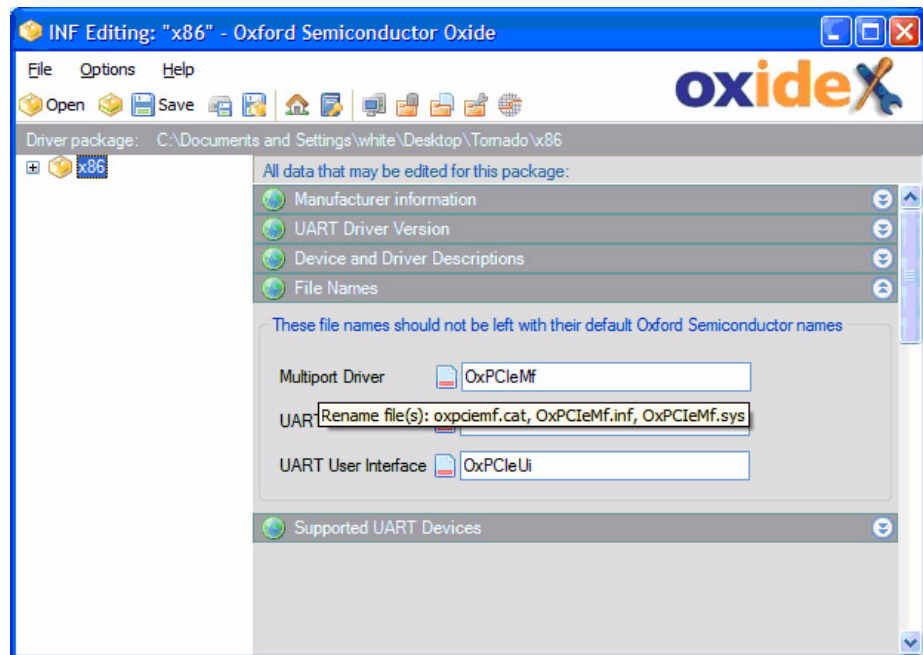
The example screen above shows part of a list of default UART device IDs supported by the PLX Technology PCI Express UART driver. The device IDs default to values determined by the configuration and mode of operation of the device. A description of the device and mode of operation is displayed next to the device ID, if it is in a recognized format.

You must update the .inf section to contain only the vendor and device IDs of the card to be supplied to end users. You must add the vendor and device IDs that you specified using the Oxide EEPROM editor, and remove any other PLX Technology vendor and device IDs. To do so, use the buttons in the .inf section as follows:

-  to delete an entry
-  to add an entry
-  to delete all entries

File Names

Many driver packages have a file name section.



To minimize the chance of naming clashes with other companies that use PLX Technology devices and drivers, it is strongly recommended that you change the default file names supplied in the PLX Technology driver package.

INF File Manipulation

This section describes how to modify .inf file contents.

Opening a Package

To work on a package, either use **File - Open** or click **Open** on the tool bar to open the **Browse for Folder** window. The INF editor opens and displays the .inf files in the selected folder.

If an INF configuration with unsaved changes is loaded when you change the device type, the INF editor prompts you to save it before proceeding.

Creating a Package

To create a new package, do one of the following:

- On the initial INF editing window, click **Create new package**
- On the INF editing main window, use **File - New**
- On the INF editing main window, click the **Create new package** button

Creating a new package involves making a copy of an existing driver package (typically the initial package we supplied). The process is:

- 1 Select the location of the existing driver package folder.
- 2 You are prompted for the location where the copy is to be made—you can create a new folder if required. It is best to use an empty folder for the destination, to avoid accidentally overwriting any existing content.
- 3 Oxide copies the entire contents of the original package to the target folder and displays the details of the .inf files.

Saving a Package

When you have completed your changes to the .inf file contents, you save the updated driver package. The options for saving the updated package are:

- **Save**—stores the package updates for the .inf files you have updated
- **Save As**—prompts for a destination folder and then makes a copy of the driver package before saving the updated .inf files into the new package

The save options are available both from the **File** menu and as tool bar buttons.

If you change file names in the driver package, the files are renamed automatically when you save the package.



Any changes made to the .inf files invalidate the digital signature applied to the default package as provided by any .cat files and including WHQL certification. For this reason, if the .inf files contain any references to .cat files when a driver package is saved, the .cat files are automatically emptied by the INF editor.

You can set Oxide to automatically recreate and digitally sign catalog files when the driver package is saved. For more information, see [Digitally Signing a Driver Package](#).

Digitally Signing a Driver Package

Changes to a driver package invalidate the catalog (*.cat) files, causing Oxide to delete the file contents. To configure Oxide to automatically recreate these, generate catalog files and digitally sign them, you must follow the steps described in [Setting up Digital Signing](#) on page 6.

When accessed using the INF editor, the signing setup windows differ slightly from those in [Setting up Digital Signing](#) on page 6, providing additional information and controls specific to catalog file operation.

Once signing setup is completed, driver signing occurs automatically whenever you save the driver package.

Creating a Final Driver Package

Because Oxide cannot edit the contents of the .inf files contained in a final driver package, we recommend that you only create the final driver package once you have finished all other development tasks. This includes initial setup (see [Initial Setup](#) on page 5), EEPROM editing (see Chapter 4 [EEPROM Editing](#)), .inf file editing (see this chapter) and branding (see Chapter 6 [Driver Branding](#)).

We recommend that you retain a master copy of the driver package, with the Oxide support content still intact.

You can create an installer using your driver package; for more information, see Chapter 7 [Installer Creation](#).

When you create a final driver package, Oxide removes from the .inf files the additional information that describes content to be edited; for more information, see [.inf File Additional Information](#) on page 59.

To create a final driver package, use **File - Create Release Package** or the equivalent tool bar button. Its operation is largely identical to the **Save As** option—Oxide prompts for a folder in which to create the driver package. When you create the updated .inf files from the information in the INF editor, all Oxide-specific additional information is removed.

Driver Branding

You use the Oxide Branding utility to brand and customize a PLX Technology driver package; you do not need to access the source code or have programming expertise. It enables you to brand your driver user interface with your company details and to modify the contents and functionality provided by the user interface and driver.

For example, in a UART driver package, you can modify:

- The logo and text displayed at the top of each customer property page
- Tabs to be displayed and tab names
- Sections to be displayed in each tab
- Default values for some properties (such as trigger levels)
- Restrictions on standard functionality (such as FIFO depth, baud rate range)

Branding Files

The driver package folder contains a branding configuration file with an extension of **.oxbx** where **x** is unique to each driver package. For example, the PCI Express UART driver package branding configuration file has extension **.oxbe**.

The branding file is copied onto the system when you install the device driver, and is read by the driver and user interface component.

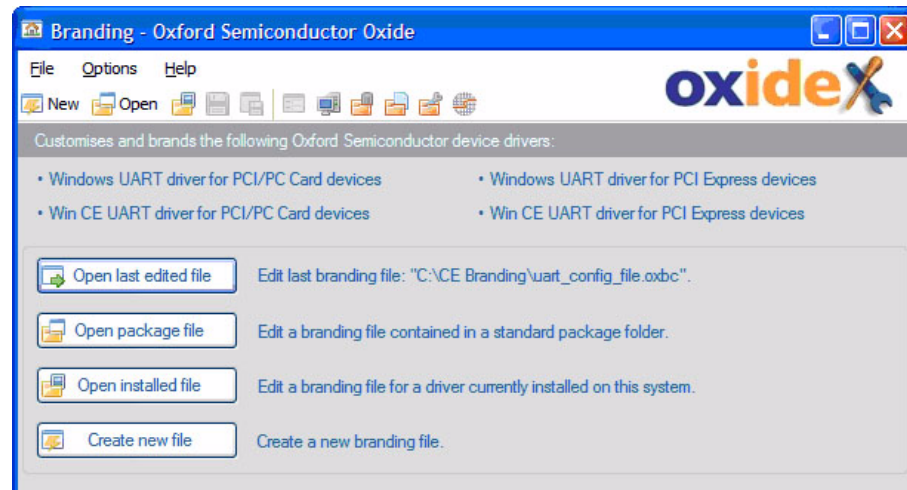


We recommend that you do not use the default branding file name on a system to be delivered to the end user. Most driver packages automatically rename this file on your behalf when edited using the INF editor. For an example of driver package file names, see [File Names](#) on page 66. Some older driver packages may have a branding file entry similar to that shown in [File Names](#) on page 66. In this case you must manually store the branding file with the name you have specified.

Starting the Driver Branding Utility

To start the driver branding utility, either select **Driver Branding** from the Windows Start menu or click **Branding** on the Oxide Launcher window.

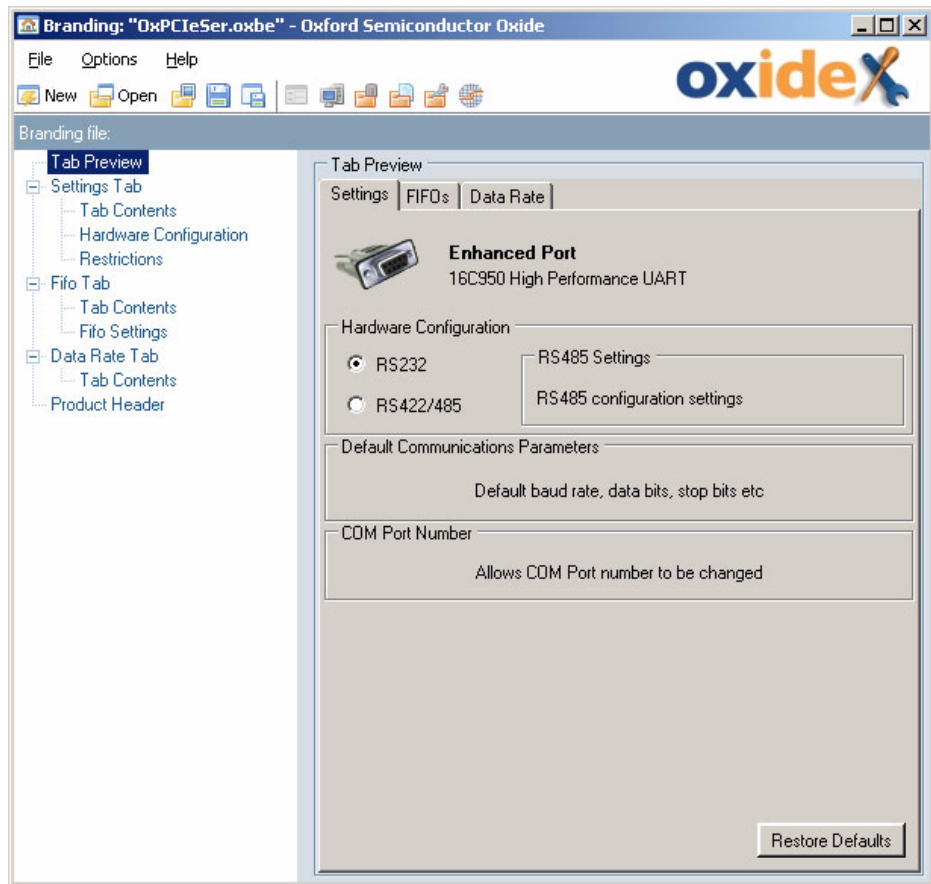
It is not mandatory for a device to be present when you run the branding utility. The initial branding window shows driver packages currently supported by the Oxide driver branding utility.



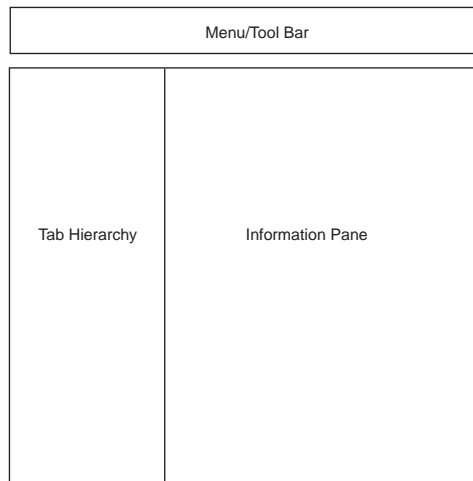
The options are:

- **Open last edited file**—not available when you first run the utility. Click to open the last branding file you edited
- **Open package file**—to select a branding file in an existing driver package. The branding file opens for you to review and modify
- **Open installed file**—to open the branding file for a driver already installed and working on the system. For more information on working with the file, see [Opening and Modifying an Installed Branding File](#) on page 76
- **Create new file**—to choose the type of branding file to create and to display the default tab preview values for the type of branding file. For more information, see [Creating a Branding File](#) on page 76

All the above options open the Branding main window.



The Branding main window is divided into panes.

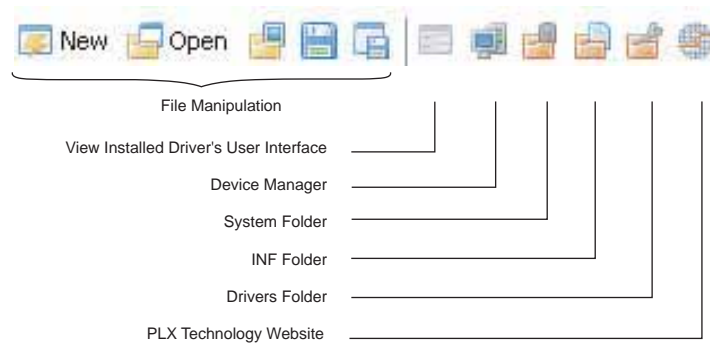


The panes display:

- Menu and tool bar—offering short cuts to common activities, as explained more fully in [Branding Tool Bar](#) on page 72
- Tab hierarchy—showing tabs and tab information for the device. For details, see [Tab Hierarchy Contents](#)
- Information pane—typically showing settings for the selected tab or tab subsection; see [Information Pane Contents](#)

Branding Tool Bar

The Branding window has the following tool bar.



Some buttons are not available if no device is attached to the bus.

The tool bar buttons are:

- Standard file handling—new branding file, open branding file, save and save as options
- Open installed file—opens the branding file for a driver already installed and working on the system; see [Opening and Modifying an Installed Branding File](#) on page 76
- View installed driver's user interface—displays the user interface of an already installed driver in Windows; see [Opening and Modifying an Installed Branding File](#) on page 76
- Facilities:
 - Device Manager— opens the Windows Device Manager
 - System folder—opens the Windows system32 folder
 - Inf folder—opens the Windows inf folder
 - Driver Folder—opens the Windows Drivers folder
 - PLX Technology Website—opens the PLX Technology website

Branding Menu Bar

The menu bar at the top of the branding main window allows you to manage the branding files:

- **File**—to manage the branding files and open a branding file in a package that has already been installed. See [File Menu](#) below
- **Options**—to navigate to useful folders; to view the interface of an installed driver using the Windows Device Manager. See [Options Menu](#) below

File Menu

It is not necessary for a device to be present on the bus to work with a branding file. The following options are available from the **File** menu:

- **Open**—to open an existing branding file; see [Opening a Branding File](#) on page 76
- **Save, Save As**—to save a modified branding file or save it as a new file
- **New**—to create a new branding file; see [Creating a Branding File](#) on page 76
- **Open installed**—to open interface details for an installed driver file. You can apply changes you make straight away or save them to a new file; see [Opening and Modifying an Installed Branding File](#) on page 76

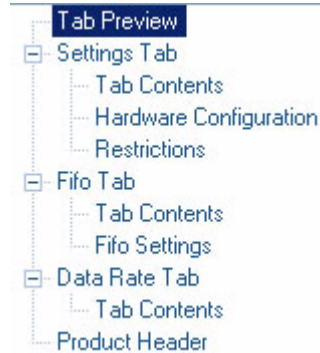
Options Menu

Using the **Options** menu, you can:

- View the properties page for a port on your system that is using an installed version of the driver being branded
- **Device Manager**—to open the Windows Device Manager
- **System Folder**—to open the Windows System 32 folder
- **Inf Folder**—to open the Windows inf folder
- **Drivers Folder**—to open the Windows Drivers folder

Tab Hierarchy Contents

The tab hierarchy controls the display in the information pane. The pane shows the tabs available for branding the chosen driver. File pane contents are initially expanded.



Tab Preview is displayed when you first open a branding file. It displays all available details in the information pane, in the same format as the Windows Device Manager.

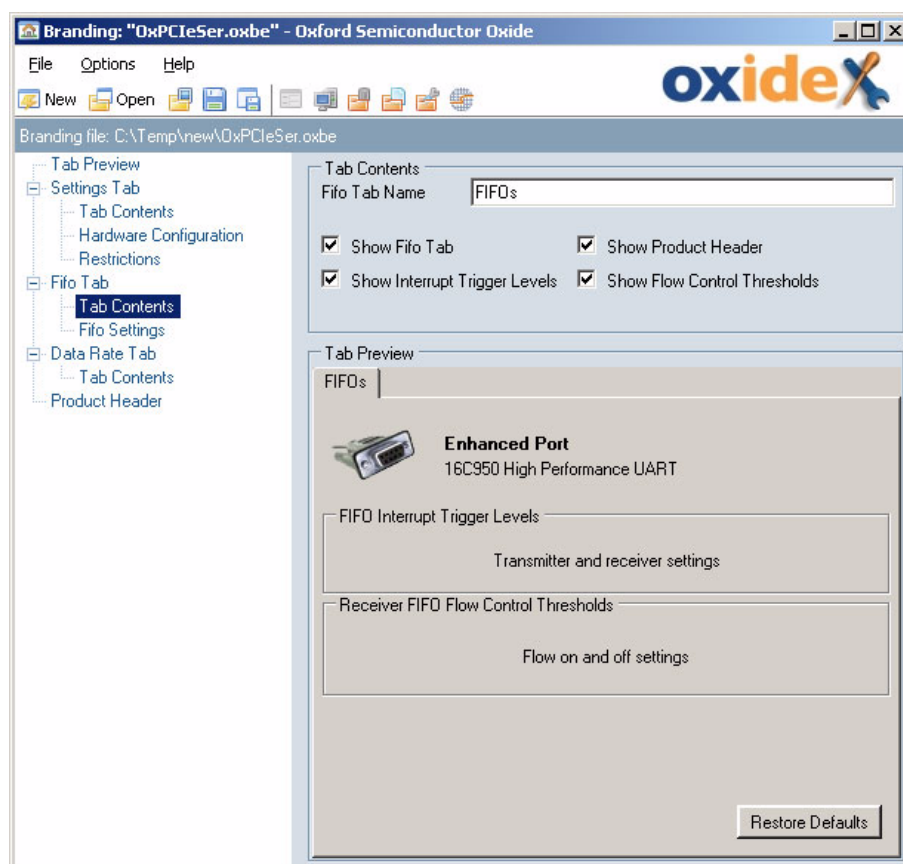
Click a tab entry to display details for that tab.

Information Pane Contents

The tab information pane displays information for the selected tab or tab subsection. A typical tab preview is similar to the user interface displayed for the device in the Device Manager properties. It shows the items that can be displayed and their location, although not necessarily full details of the controls.

When you make a change, the Tab Preview pane shows how the tab appears in the Windows Device Manager.

The first lower level entry for a tab is typically **Tab Contents**. You use this to select the items in the user interface you want to display. The screen below shows typical **FIFO Tab** tab contents for a PCI Express UART device driver.



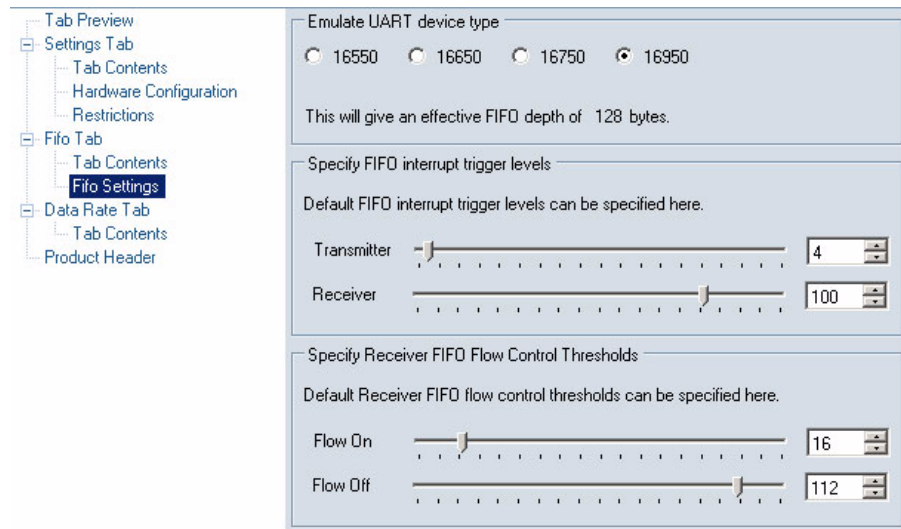
You use the **Tab Contents** section to specify the items shown in the remaining tab subsections by checking and unchecking the check boxes. You can even suppress the tab display completely, if required.

You can use the **Product Header** tab to define a graphic to display and to modify the text displayed on the tab.



By unchecking a check box that controls whether information is displayed in the tab, you prevent it appearing in the tab display. However, you may need to specify the settings that still apply. Using the above screen as an example, if you uncheck the box **Show Interrupt Trigger Levels** to prevent users changing the trigger levels, you still need to specify what the default trigger levels are. To do so, use the **FIFO Settings** tab.

To display device settings in the information pane, select one of the lower level tabs in the tab hierarchy. For example, The screen below shows a typical **FIFO Settings** tab section for the PCI Express device driver.



Opening a Branding File

To work on a branding file, either use **File - Open** or click the **Open** button to open the Open UI Branding File window. Browse to the branding file you want.



If a branding file is loaded when you change the branding file, the branding editor prompts you to save it before proceeding.

Creating a Branding File

To create a new branding file, use **File - New**, click **New** on the tool bar or click **Create new file** on the initial Branding window (see [Starting the Driver Branding Utility](#) on page 70). You are asked to select the type of branding file you want to create (for example, PCI Express UART driver). The Branding window displays default values for the type you selected. After editing the file, you are prompted to save it.

Opening and Modifying an Installed Branding File

To customize a branding file for a device driver that is already installed, do one of:

- Click **Open installed file** on the initial Branding window
- Use **File - Open Installed**
- Click **Open installed** on the tool bar. The main Branding window shows details for the installed driver

When you save the details, they are immediately applied to the user interface for the driver. You can see the effect by using **Options - View Driver User Interface**, which opens the driver property page. Alternatively, you can save the settings to a new branding file for use with a different driver package.



To view the current user interface details if no modification is necessary, use **Options - View Driver User Interface**.

Installer Creation

The Oxide Installer Creation utility enables you to create a standalone installer for one or more Windows device drivers.

Normally when a new card is inserted, Windows prompts you to select the driver package to install. However, when redistributing drivers to an end user, it is often more convenient to pre-install these drivers so that they are immediately available when the card is inserted.

Before you create an installer, you must:

- Obtain Microsoft WHQL certification, if you want it (see [Setting up Digital Signing](#) on page 6)
- Create the final driver package (see Chapter 5 [INF Editing](#))

Different applications may have different requirements of an installer, so the utility provides several delivery options:

- A folder containing a Setup.exe file which starts the installation, along with separate driver and support files
- A zip file containing the same content as the above folder
- A single executable file which contains all the drivers and support information, and starts the installation when run

The installer is based on Microsoft dpinst installer technology. This provides several options to control the installation. These include:

- Whether to display a user interface during installation
- Whether to display a license agreement
- Customized graphics and text in any user interface
- Specifying text in several languages

Once you have created the installer, you can specify additional optional command line parameters.

Starting Installer Creation

To run the Installer Creation utility, either go to the **Start** menu and click the shortcut, or go to the Launcher window and click **Installer Creation** (see [Running Oxide and Launching a Utility](#) on page 3). The Installer Creation initial window is shown.



You can open and edit any format installer you created, whether folder, zip file or executable. The options are:

- **Open last installer**—to reopen the last used installer
- **Open installer**—to browse to and open an installer



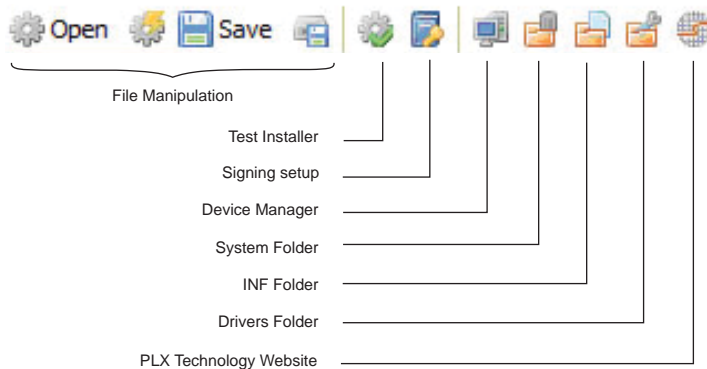
To use dpinst installer technology with installers that are in folder format, you must have a driver folder that contains a file called **dpinst.xml**. To open the folder format driver, you open dpinst.xml.

- **Create new installer**—to create an empty installer containing default settings, graphics and text

The Installation Creation main window is shown, see [Setting up Installer Creation](#) on page 80.

Installer Creation Tool Bar

The Installer Creation window tool bar is shown below.



The tool bar buttons are:

- Standard file handling—new installer file, open installer file, save and save as options
- Test installer—tests the installer you create
- Signing setup—opens the Signing Setup window, so you can check the entries
- Facilities:
 - Device Manager—opens the Windows Device Manager
 - System folder—opens the Windows system32 folder
 - Inf folder—opens the Windows inf folder
 - Driver Folder—opens the Windows Drivers folder
 - PLX Technology Website—opens the PLX Technology website

Installer Creation Menu Bar

The menu bar at the top of the Installer Creation main window allows you to control installer creation activity:

- **File**—to manage installer files; see [File Menu](#)
- **Options**—to configure Installer Creation settings; see [Options Menu](#) on page 79

File Menu

The **File** menu options are:

- **Open**—to open an installer file
- **New**—to create a new installer file
- **Save**—to save an installer file
- **Save As**—to save an installer file in a location you choose

Options Menu

Use the **Options** menu to access information and test the installer:

- **Test Installer**—to test the installer file; for more information, see [Testing an Installer](#) on page 90
- **Signing Setup**—to set up digital signing; for more information, see [Setting up Digital Signing](#) on page 6
- **Device Manager**—to open the Windows Device Manager
- **System Folder**—to open the Windows system32 folder
- **Inf Folder**—to open the Windows inf folder
- **Drivers Folder**—to open the Windows Drivers folder

File Signing

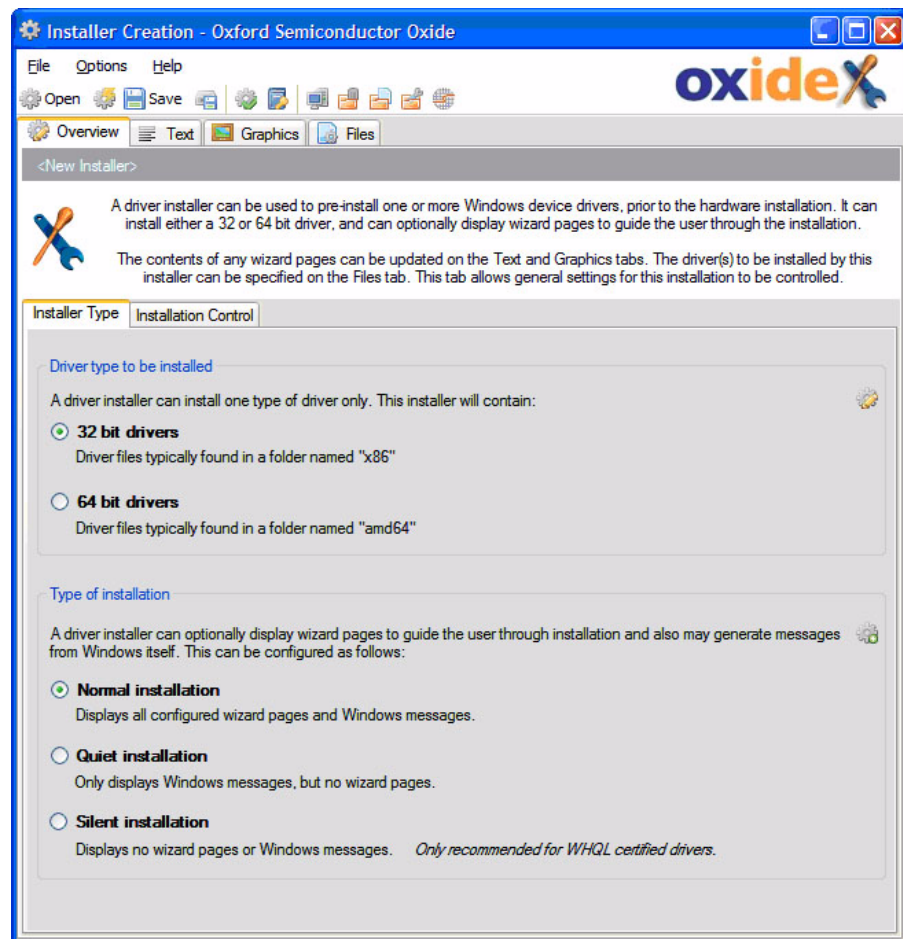
If you have set up file signing in Oxide (see [Setting up Digital Signing](#) on page 6), then executables created by this utility are automatically digitally signed using your digital certificate. This is particularly important on later Windows versions, such as Vista.

The signing does NOT apply to the contents of a driver package. The driver is typically signed by the INF editor (see Chapter 5 [INF Editing](#)). If you have Microsoft WHQL certification for your driver, it is important that the installer does not modify the driver signature provided by Microsoft.

To check the signing setup from within the Installer Creation utility, use the **Options** menu or equivalent tool bar button.

Setting up Installer Creation

The Installer Creation main window has tabs for different categories of information: installation overview (with Installer Type and Installation Control secondary tabs), text to be displayed, graphics to be displayed and files to be installed.



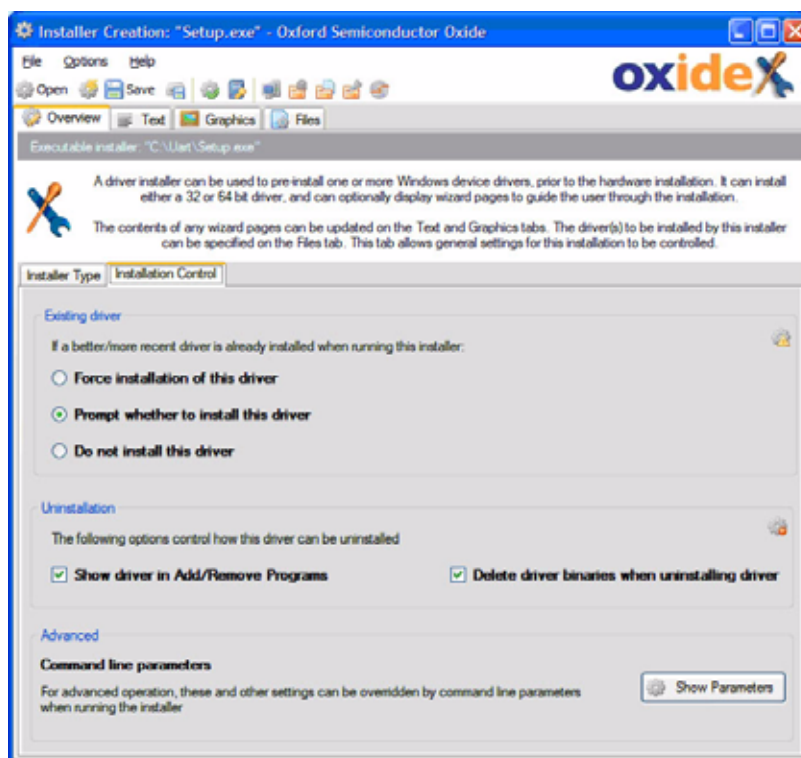
To set installer type details:

- 1 On the Installer Type tab, under **Driver type to be installed**, click the radio button for the bit size you want. A driver installer must install only 32 or 64 bit drivers: you must provide separate installer packages for 32 and 64 bit operating systems. The installer package is created with a 32 or 64 bit setup.exe, depending on which option you select.

- 2 To set the types of messages displayed during installation, under **Type of Installation**, click the relevant radio button. Message types include messages displayed by the installer (as specified and customized using this utility), and messages generated by Windows. To control which are displayed, choose one of:
 - Normal Installation**—displays both installer and Windows messages
 - Quiet Installation**—displays only Windows messages. No settings in the Text or Graphics tab need be configured as these have no effect
 - Silent Installation**—displays no messages. No settings in the Text or Graphics tab need be configured as these have no effect. Select this if you are only installing fully WHQL-certified drivers. If you select this option, but Windows needs to display a message (e.g. that the driver is not WHQL certified), then the driver installation fails

To set installation and uninstallation details:

- 1 Click the Installation Control tab to display it.



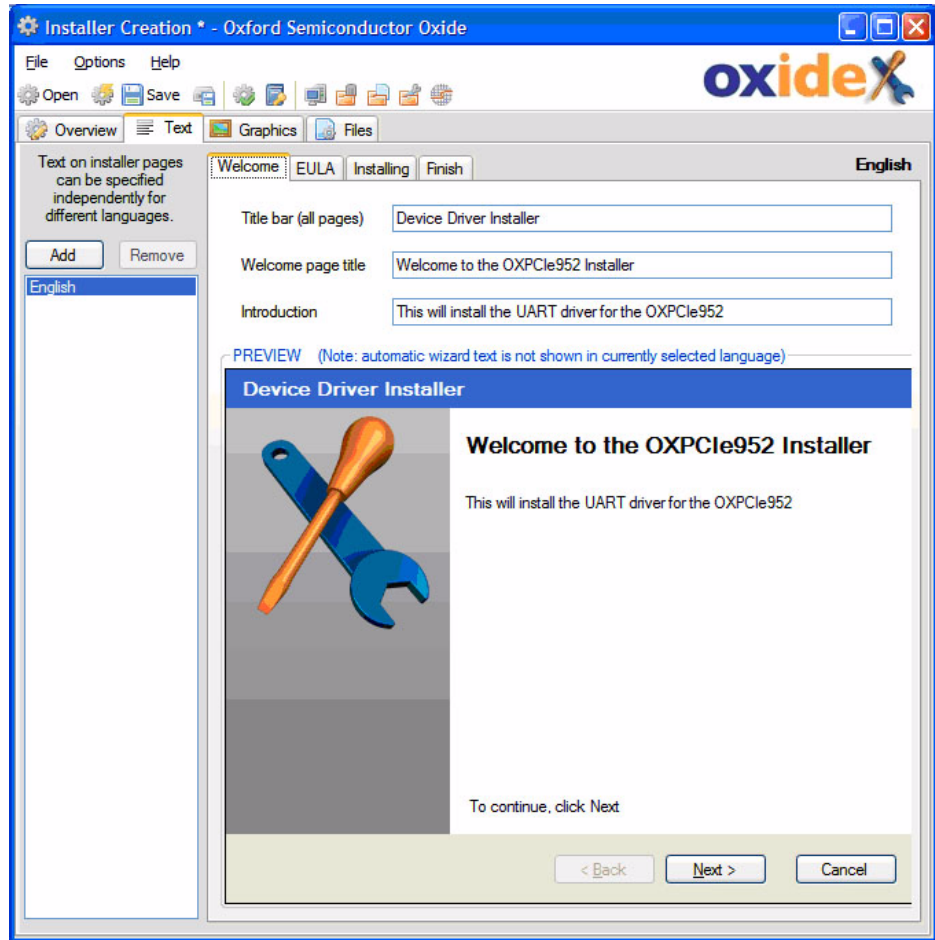
- 2 When installing a driver, Windows checks if a driver is already installed for the device, and if so whether the installed driver is better or more recent. By selecting a radio button under **Existing driver**, you can control what happens if such a driver does exist: you can ask the user which driver to use, force the installation of this driver, or not install the driver so that the already installed one is used.
- 3 To make this driver appears in the Windows Add/Remove Programs list, check **Show driver in Add/Remove Programs**. To allow the binaries to remain on the system when uninstalling, do not check **Delete driver binaries** (for advanced use only).
- 4 You can view information on the parameters that you can specify on the setup command line to modify or override the installation settings. For more information on the parameters, see [Command Line Parameters](#) on page 90.

Defining Installer Text

The installer user interface can display the following pages during the installation:

- A Welcome page, providing an overview of what is being installed
- An optional End User License Agreement (EULA), which must be accepted for the installation to continue
- An Installing page showing the progress through the installation
- A Finish page when the installation has completed

You can customize the text on each of these pages using the secondary tabs on the **Text** tab.



You can define the text for each of the following languages:

- | | | |
|------------------|-----------|-----------------------|
| Arabic | French | Norwegian |
| Chinese (China) | German | Polish |
| Chinese (Taiwan) | Greek | Portuguese (Brazil) |
| Czech | Hebrew | Portuguese (Portugal) |
| Danish | Hungarian | Russian |
| Dutch | Italian | Spanish |
| English | Japanese | Swedish |
| Finnish | Korean | Turkish |

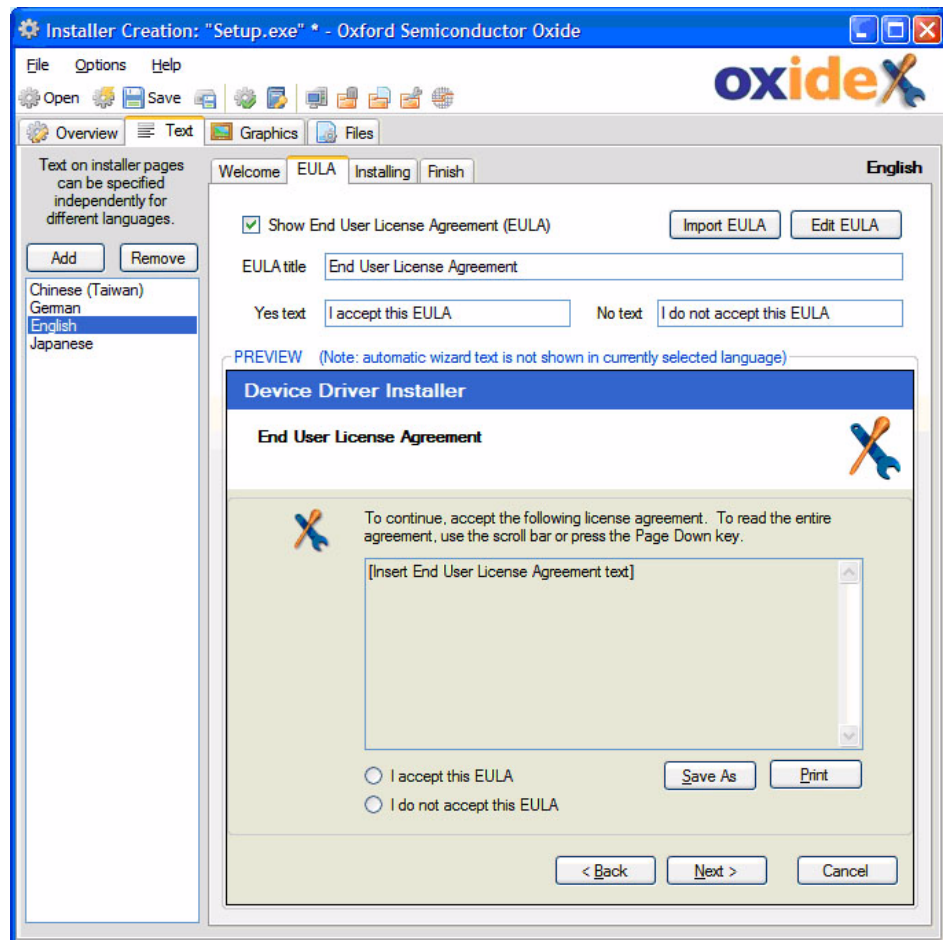
You must ensure you have all the correct fonts and input methods installed on your PC for the languages you are specifying in order to obtain the expected results. This is particularly important for right-to-left languages (such as Arabic and Hebrew).

When installing, the installer chooses the language most appropriate for the current system.

You can add and remove languages using the **Add** and **Remove** buttons on the left on the window.

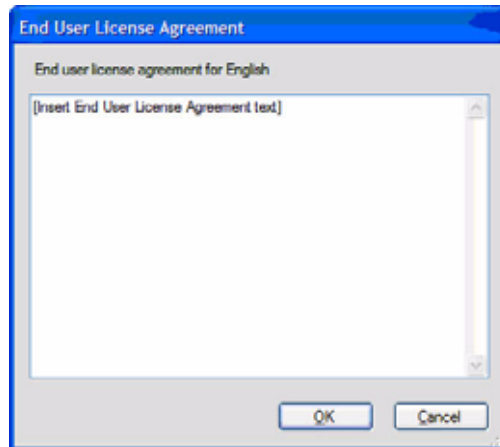
To see and modify the text for a language, select the language in the list on the left. The text for each tab is shown in two parts: controls where you modify the text, and an approximate preview of the page displayed by the installer. Text that the installer generates (e.g. "To continue, click Next") is always shown in English in the preview. When installing the driver, this is automatically translated into the correct language.

The Installing and Finish tabs are similar to the Welcome tab. However, the EULA tab has some additional settings.



The EULA page is optional, and can be enabled, disabled and configured independently for each language. You can import the contents of the EULA from a TXT file, by clicking **Import EULA**. You can use a Unicode encoded text file, so it supports EULA text in any language.

Alternately you can enter the EULA yourself, or post-edit an imported EULA by clicking **Edit EULA**, which shows the EULA editing window.



Specifying Installer Graphics

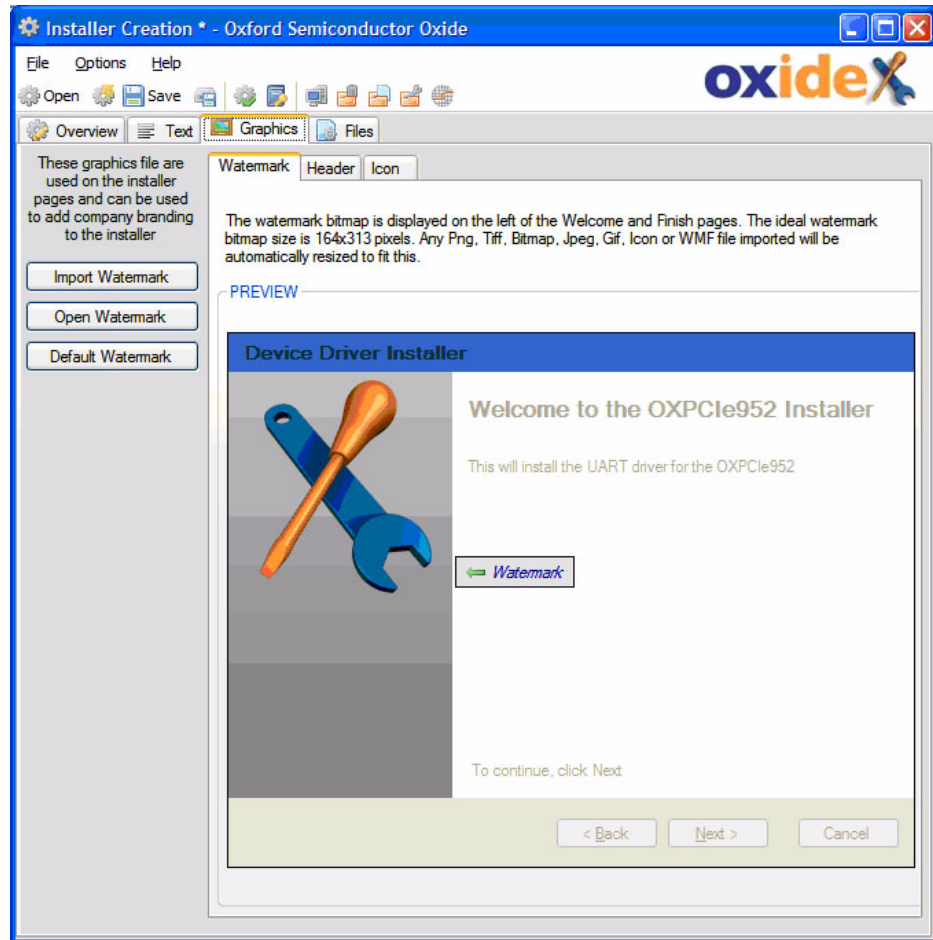
You can customize each installer page by adding graphics. Oxide includes default graphics, but you can import different files.

To do so, on the Installer Creation window, click the Graphics tab to display it. Use the buttons on the left hand of the window to specify your graphics. The options are:

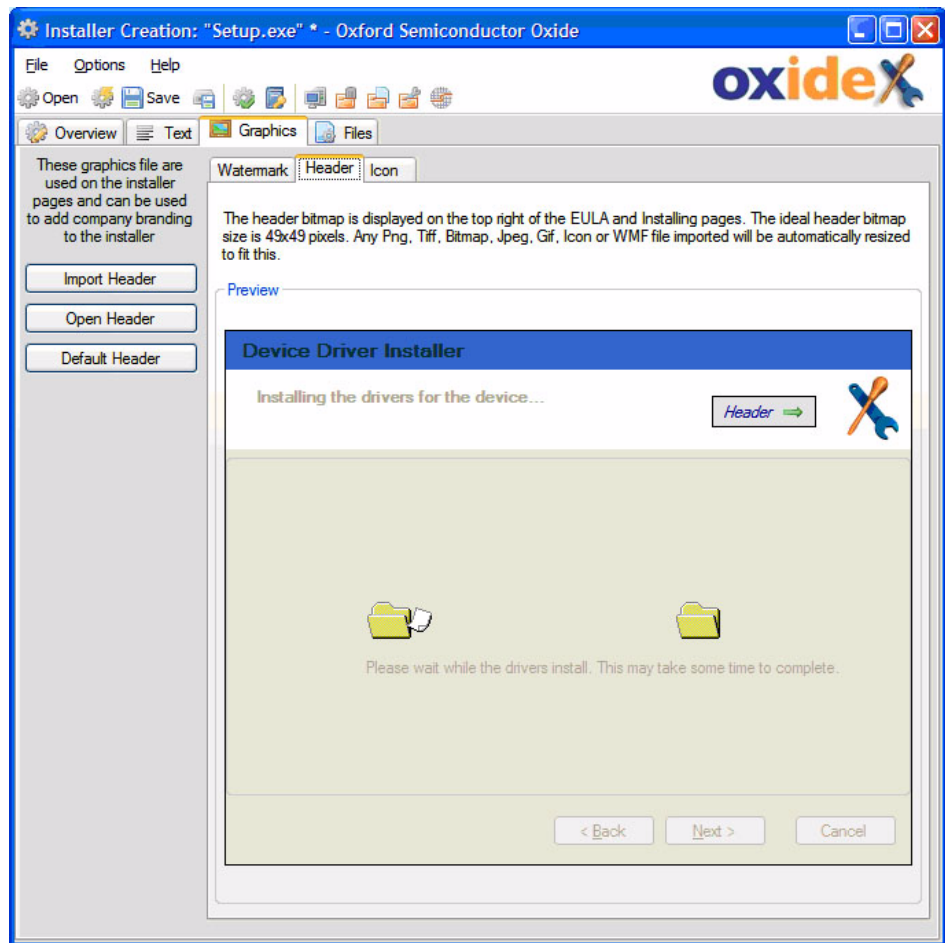
- **Import**—to import a graphic file
- **Open**—to open and edit the selected graphic in the default graphics application on your PC
- **Default**—to set this graphic back to the default provided by Oxide

The supported graphics are:

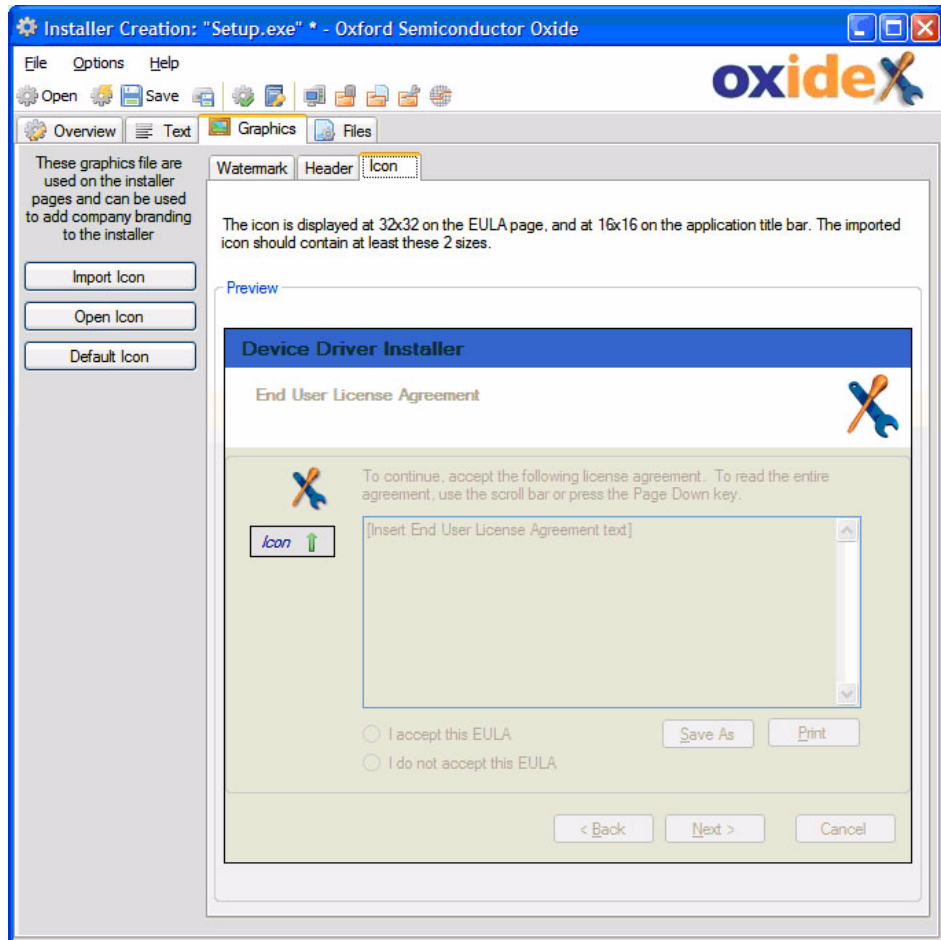
- **Watermark**—ideal size is 164 x 313 pixels, and can be imported from a png, tiff, bitmap, jpeg, gif, icon or wmf file.



- **Header**—ideal size is 49 x 49 pixels, and can be imported from a png, tiff, bitmap, jpeg, gif, icon or wmf file.

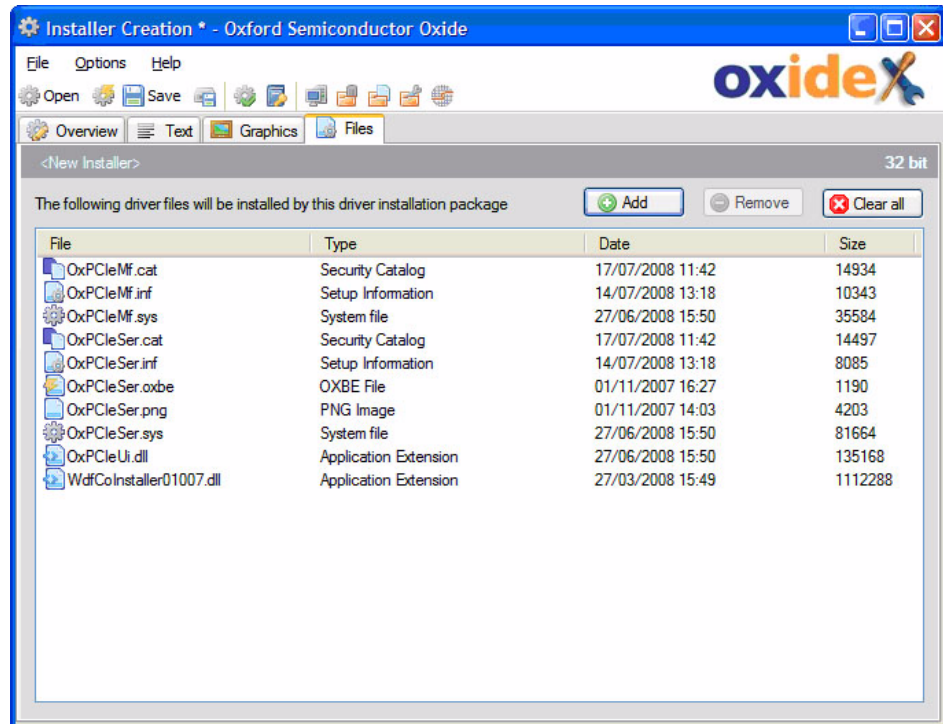


- **Icon**—this is used at 32 x 32 pixels on the EULA page, and 16 x 16 pixels on the tool bar. It can also be seen in Windows Explorer as an executable icon. You can import it from an icon file.



Specifying Driver Files

You specify the driver files to be installed on the Files tab.



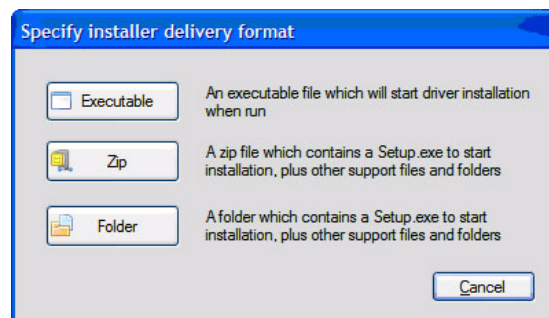
To specify the driver files, click **Add** and browse to the folder containing the driver files (the folder level that contains the INF files, typically in the **x86** folder for 32 bit drivers or **amd64** folder for 64 bit drivers). All contents of the folder are added.

To install more than one driver using the same installer package, click **Add** again and browse to another folder containing the driver files you want. Select the second driver to be added: you are given the option of either merging the new driver files into the existing content, or replacing all the files. To produce a combined installer package, merge the files. Do not mix 32 and 64 bit drivers in the same installer.

To remove all the current files in the installer, click **Clear all**. To delete individual files from this list, select a file and click **Remove**. This can be useful if there were files in the driver folder that should not be installed as part of the driver.

Saving an Installer

When saving an installer for the first time, or when you select **Save As**, you are prompted to select the type of installer to save.



You can use **Save As** to store an opened installer in a different form.

Testing an Installer

You must perform the final installation test using the saved installer package. However, Oxide also provides a quick method for testing the installer from within the Installer Creation utility.

To test the installer, use **Options - Test Installer** or the equivalent tool bar button. The installer runs and installs the drivers on your test system.

To see how the installation looks in a specific language rather than the default language on your PC, click the Text tab and select the language. Next use **Options - Test Installer** or the equivalent tool bar button to see the installation using the selected language.

Command Line Parameters

You can override the default settings in an installer by using command line parameters. This may be useful in advanced operation, such as in an installation script. You open a command line window for a saved installer using the executable created, or if you saved it as a folder or zip file, by running the Setup.exe that it contains.

To add the parameters to an installer in executable format, add **-ap "<parameters>"** to the command line.

To add the parameters to a Setup.exe in an installer folder or zip file, enter just the parameters on the command line.



For further information on these parameters and advanced installer configuration options, see the online documentation for the Microsoft utility **DPIInst.exe**.

You can view information for the following parameters when you click **Show Parameters** on the Overview, Installation Control tab.

Parameter	Description
/U <path to INF file>	Uninstalls a driver package (.inf file)
/S /Q	Suppresses the device installation wizard and any windows opened by the operating system
/LM	Legacy mode. Accepts unsigned driver packages and packages with missing files. Such packages may not install on the later versions of Windows
/P	Prompts if the driver package to be installed is not better than the current one
/F	Force installs if the driver package is not better than the current one
/SH	Scans hardware for matching devices and only copies and installs those drivers for which a device is present. Plug and Play drivers only
/SA	Suppresses the Add/Remove Programs entry normally created for each driver package
/A	Installs all or none
/EL	Enables all languages not explicitly listed in the installer
/L LanguageID	Tries to use the given language in all UI. Useful for localization tests
/SE	Suppresses the EULA
/D	Deletes driver binaries on uninstall
/SW	Suppresses the device installation wizard, the operating system may still open windows

This page is intentionally blank

OxShell

The PLX Technology OxShell utility enables you to modify the EEPROM contents of PLX Technology bridge devices. It is ideal for modifying EEPROM contents in the field and for troubleshooting.

For more information about the Oxide EEPROM editor, see Chapter 4 [EEPROM Editing](#).

When you install Oxide, a command prompt shortcut suitable for running OxShell is installed in the **Start** menu.

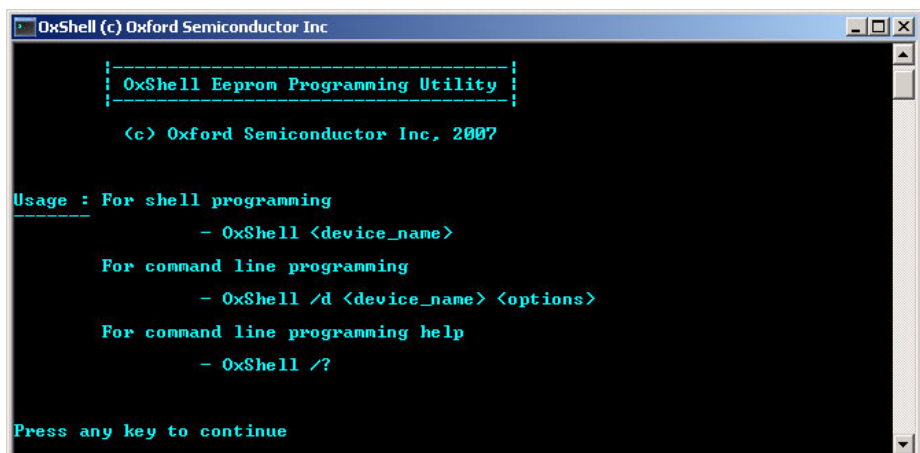
Running OxShell

OxShell provides the following modes of programming a device EEPROM:

- As a command shell, in which you can execute commands to read, write, edit and compare EEPROM data
- As a command line tool for production programming; it can also be used from within scripts to automate operation

You open either mode from a Windows command line prompt in a command window. To open a command window, go to your **Start** menu and click OxShell.

To display an explanation of how to run OxShell in the mode you want, type OxShell at the command prompt. The OxShell usage screen is shown.



```
OxShell (c) Oxford Semiconductor Inc

OxShell Eeprom Programming Utility

<c> Oxford Semiconductor Inc, 2007

Usage : For shell programming
- OxShell <device_name>

For command line programming
- OxShell /d <device_name> <options>

For command line programming help
- OxShell /?

Press any key to continue
```

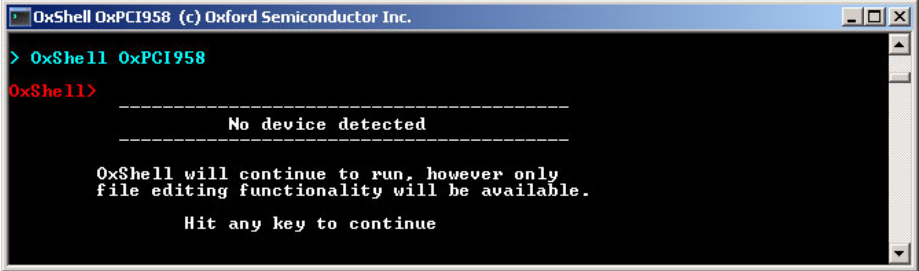
Using the Command Shell Tool

To run OxShell as a command shell, at the command prompt type:

OxShell <device>

where <device> is the device attached to your host PC, for example OXCFU950.

If no device is attached to the host PC, OxShell displays the OxShell prompt and a message indicating only limited functionality is available.



```
OxShell OxPCI958 (c) Oxford Semiconductor Inc.
> OxShell OxPCI958
OxShell>
-----
                No device detected
-----
OxShell will continue to run, however only
file editing functionality will be available.
Hit any key to continue
```

If you press any key when no device is attached, you still enter the command shell, but all commands that require a device and an EEPROM are disabled. You can still edit, verify and save images.

If a device is detected, the OxShell command prompt is displayed.



```
OxShell OxPCI958 (c) Oxford Semiconductor Inc.
OxShell>
```

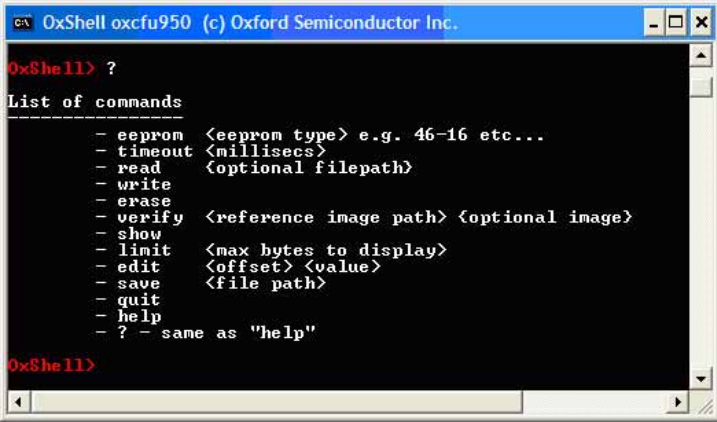
You can load and save files in binary and hexadecimal format. The hexadecimal format is specific to PLX Technology. Although you can edit a file outside OxShell and then reload it, you must not change the line endings of a hexadecimal file.

Hexadecimal files have the extension **.oxx** and binary files have the extension **.oxi**.

For information on the commands and parameters you can enter, see [Commands](#) on page 96. In the command shell window, you can cycle through previous commands, edit and resubmit them.

Viewing and Printing Command Shell Help

To view information about the commands you can enter, type `?` or `/?` at the OxShell prompt. The command shell help screen is displayed. The commands are detailed in [Commands](#) on page 96.



```
OxShell oxcfu950 (c) Oxford Semiconductor Inc.
OxShell> ?
List of commands
-----
- eeprom <eeprom type> e.g. 46-16 etc...
- timeout <milliseconds>
- read <optional filepath>
- write
- erase
- verify <reference image path> <optional image>
- show
- limit <max bytes to display>
- edit <offset> <value>
- save <file path>
- quit
- help
- ? - same as "help"
OxShell>
```

To print help information on the commands, type **Help** at the OxShell prompt.

Closing OxShell

To close OxShell:

- 1 At the OxShell prompt, type **Quit** or **Exit** to terminate the program and return to the command window.
- 2 Type **Exit** to close the command window.

Using the Command Line Tool

When you run OxShell as a command line application for EEPROM reading and writing, you provide all the parameters you want, then Oxide completes the commands without further user input. This mode of use is intended for automated environments where OxShell is either called in a batch file or spawned by another program to provide unattended EEPROM programming and verification.

To run OxShell as a command line tool, at the command prompt type:

```
OxShell /d <device>
```

where *<device>* is the device attached to your host PC, for example OXCFU950.

Command Examples

To verify the device contents against an image file, type:

```
OxShell /d <device> /v <image file>.oxi
```

where *<image file>.oxi* is the name of an image file, such as **test1.oxi**.

To erase the contents of the device EEPROM, type:

```
OxShell /d <device> /e
```

Automated Command Examples

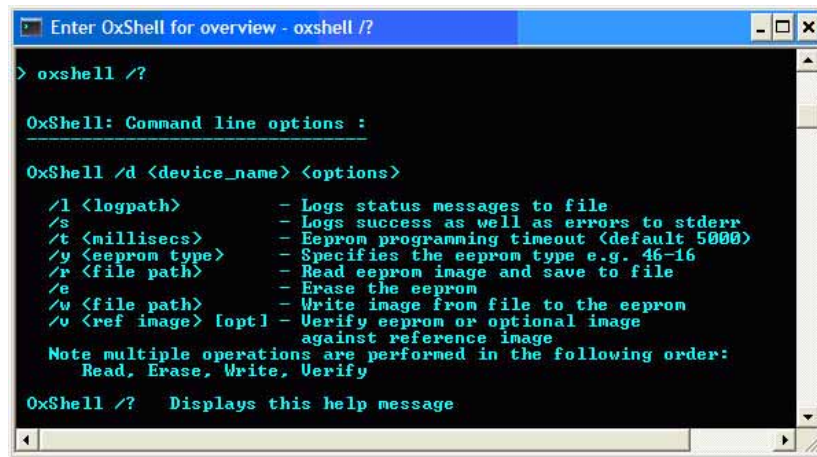
For automated running, the following commands are typical:

- OxShell /d OxCFU950 /t 5000 /l log.txt /w binary_image.oxi
- OxShell /d OxCFU950 /l log.txt /v binary_image.oxi

For more information on these commands, see [Commands](#) on page 96.

Viewing Command Line Help

To view information about the commands you can enter, type */?* at the OxShell prompt. The command line help screen is displayed. The commands are detailed in [Commands](#) on page 96.



Commands

[Table 8-1](#) summarizes the OxShell commands, parameters and their actions, for the command shell and the command line.

You must surround any parameters that include spaces by quotes; for example: **verify "my reference file.oxi"**. Otherwise, parameters do not need quotes.

Command Shell	Command Line	Parameters	Action
?	/?		Displays the command set with parameters and waits until you press any key; see Viewing and Printing Command Shell Help on page 95 and Viewing Command Line Help on page 96
	/l	<logfile path>	Logs errors and successes to <logfile path> (see Error Messages on page 99 for details). New errors are appended to an existing file, otherwise <logfile path> is created. We recommend you to specify a different log file for each device type, because the log file does not record the device type
	/s		Logs errors and successes to stderr instead of to a file. Do not use at the same time as /l

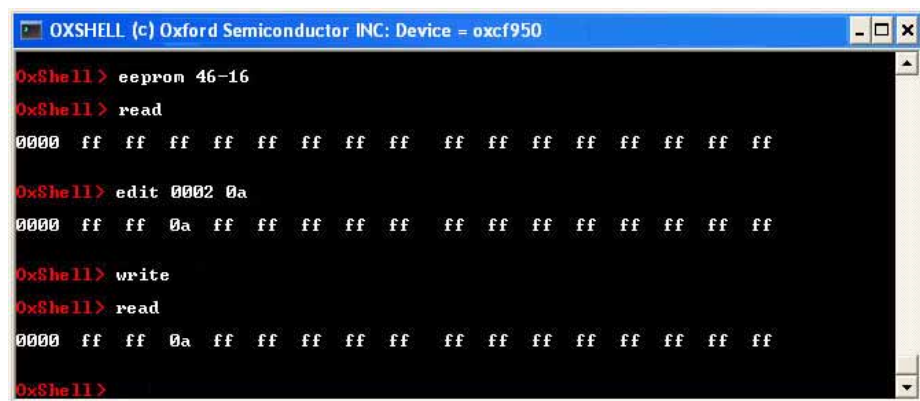
Table 8-1 OxShell Commands and Parameters (Sheet 2 of 3)			
Command Shell	Command Line	Parameters	Action
timeout	/t	<millisecs>	Applies the <millisecs> timeout to EEPROM programming operations
verify	/v	<reference image> [optional image]	Verifies the contents loaded from the EEPROM against the supplied reference file image. If two images are supplied, verifies [optional image] against <reference image>. The comparison result is displayed on the screen until you press any key. The EEPROM or optional image data is parsed to the same format as the reference image before comparing. Do not use with /e
erase	/e		Erases the EEPROM by writing 1s to it. Do not use with /v
	/d	<device>	Selects the device currently attached to the host PC and whose EEPROM is to be used
read	/r	<file path>	Reads and displays for editing the contents of <file path>, if supplied, otherwise reads and displays the EEPROM. Binary or basic formats are acceptable. The edited file can be reprogrammed into the EEPROM or saved back to <file path>
show			Displays the image that is currently stored in memory. If no image is loaded, OxShell displays a warning
edit		<offset> <value>	Sets the byte in EEPROM memory at <offset> to <value> and displays the prospective new EEPROM contents on screen. Note: Does not commit to EEPROM yet—multiple edits can be performed before writing them to the EEPROM
save		<file path>	Saves the EEPROM contents shown on screen to a named binary file. If the file exists, prompts for permission to overwrite
write	/w		Writes the current image in system memory to the EEPROM. An image must be read into system memory before it can be written to the EEPROM. On the command line, the /w switch takes a file path as an argument, but the shell write command takes no arguments
quit			Closes the application
help			Prints help on the commands
limit		<max bytes to display>	Limits the number of bytes displayed on the screen to the size given; otherwise defaults to the size of the EEPROM

Command Shell	Command Line	Parameters	Action
EEPROM	/y	46-16, 56-16, 66-16, 76-16, 86-16 (Typing -16 is optional; for example, you can just type 46)	Specifies the microwire EEPROM type. If you do not enter a parameter when in the OxShell command shell, OxShell displays the EEPROM currently configured. The OxShell command shell remembers the last EEPROM type configured and displays it when you start OxShell

Command Sequence Example

The screen below shows an example OxShell editing sequence using the command shell. The sequence loads a blank EEPROM, displays the first 16 bytes, sets the value of the byte at offset 2 to 0x0A, writes the edited value back to the EEPROM, then re-displays the EEPROM contents.

The screen shows what you see when you use the limit command to display only the first 16 bytes of each line.



Editing Keys

Table 8-2 lists keys on your keyboard to use when editing.

Key	Function
Backspace	Deletes previous character
Del	Deletes next character
Ins	Toggles between insert and overwrite modes. Cursor becomes a block shape in overwrite mode
← and →	Navigates the current edit line
↑ and ↓	Replaces the current edit line with commands from a stored command history
Home	Moves the cursor to the start of the current edit line
End	Moves the cursor to the end of the current edit line

Error Messages

When OxShell is run as a command shell, error messages are displayed on the screen.

When OxShell is run as a command line utility, it sends progress and feedback messages to **stdout** and error messages to **stderr**. You can use command **/s** to send success messages to **stderr**. You can use command **/l** to save error and success messages to a named file. For more information on these commands, see [Commands](#) on page 96.

Because OxShell command line error messages are intended for logging purposes and parsing by tools to produce reports, they follow a format designed to make parsing easy. Each logged message appears on a separate line. The format of each line is:

```
<DD/MM/YYYY>_<HH:MM:SS>_<error number>_<single word error name>_<optional supplementary information><\r\n>
```

where `_` is a single space and `\r\n` indicates the end of a line.

You can parse this format into:

- The first four space separated fields, which are always present
- Any optional supplementary text, which is whatever text and white space remains on the line after extracting the first four fields.

So for automated running, you can set OxShell to log the error/success messages into a named file which you later parse using a simple utility to create an audit log in a database.

Table 8-3 lists the errors produced when OxShell is used as a command line tool.

Table 8-3 OxShell Command Line Error Messages			
Error Code	Error Name	Supplementary Information	Comments
0	NO_ERROR		Program has been successful
1	TIMEOUT	The entire operation did not complete within the specified timeout	Timeouts are specified as a command line parameter in milliseconds. An error is issued if EEPROM programming fails to complete within this time
2	LOGFILE_FAILURE	May have file system error information attached	General error while creating, opening or writing the log file
3	UNSPECIFIED_ERROR	May have system error information attached	Unforeseen error
4	CONFIG_FAILURE	May have extra error information attached	Failure opening, reading or parsing the configuration file
5	PROGRAMMING_FAILURE		EEPROM programming has failed
6	VERIFICATION_FAILURE		EEPROM does not verify against an image, or files do not verify
7	ERASING_FAILURE		Failure trying to erase the EEPROM
8	EEPROM_READ_FAILURE		Failure trying to read the EEPROM
9	NO_DEVICE_FOUND		Cannot find the specified device
10	PARAMETER_ERROR	Usually has supplementary information	Command line parameter error

Using OxShell with the Oxide EEPROM Editing Utility

OxShell can work hand-in-hand with the Oxide EEPROM editor. Any image you create using Oxide can be used by OxShell. The golden images provided as part of Oxide are the best starting point for your more advanced custom EEPROM image requirements.

If you have an EEPROM image requirement that you cannot create using the Oxide EEPROM Editing utility, you can:

- 1 Use the EEPROM editor to save an image for the device, modifying the fields available as required.
- 2 Load the created image into OxShell.
- 3 Use the edit command to add your custom requirements.



Once you have customized an EEPROM image using the OxShell edit command, do not load this back into the EEPROM editor. It is not guaranteed that any custom changes are retained if you do. You must use this modified image only in OxShell.