Appendix—Platform-Specific Programming

Omnis Studio is a cross-platform application development tool and the vast majority of features within the environment and your finished applications will work on all platforms. However there are some issues that may affect one platform only that you may need to consider, especially if you are deploying your application to a multi-platform environment.

Major platform-specific features, such as Automation, DDE or Apple Events, are described in their own chapters in the *Extending Omnis* manual.

Windows Vista

Omnis Studio 5.0 and above fully supports Windows Vista for Development and Runtime versions. Please read the following sections for details regarding Vista support in Omnis Studio.

End User Security

Windows Vista introduced a system called *User Account Control* (UAC) to improve security against unwanted attacks of one form or another on an end-user's computer. UAC provides a secure mechanism for running end-user accounts with Standard User privileges, while removing the need for Administrator privileges when performing many common tasks, such as installing a printer driver. In Omnis, the measures introduced under UAC affect where files in the Omnis Studio product tree are stored on your computer to allow you to create/modify Omnis libraries or write to other files in your application. This will also affect how you construct installers for deploying your own products under Windows Vista.

Program and Data directories

Due to the UAC mechanism, Omnis uses two locations for installing the Omnis executable, plus any read-only files and other files that require write access during development. The Omnis installer places all files in the correct location during installation, and changes to the product tree are made automatically the first time Omnis is run.

- ❑ **Program Directory**
  For the Omnis executable and read-only files: "C:\Program Files\TigerLogic\Omnis

Studio xxx”. Note that executable files do not include Omnis libraries, as these often need write access.

- ❑ **Data Directory**
  For writeable files, which are placed in the \AppData\Local\ folder as follows (note the AppData folder is usually hidden so you may need to show it using the Tools>>Folder Options option in the Windows File Manager):

  The location of the Data Directory is either:
  C:\users\name\AppData\Local\TigerLogic\Omnis Studio xxx
  Or
  C:\Windows\System32\config\systemprofile\AppData\Local\TigerLogic\Omnis Studio xxx (when Omnis is running as a service)

The Omnis installer places all the writeable files into a folder called ‘firstruninstall’ in the Program directory: these files are then copied to the Data directory the first time Omnis Studio is run. You will need to take this into account when you are building installers for your own products, see below.

Note that Omnis can still run from a single location, provided that the location is not in Program Files, and you have write access to the tree in that location.

Under Vista, sys(115) returns the path of the Data directory, i.e. the path to all writeable files, while sys(215) returns the path of the Program directory; on all other platforms, sys(215) returns the same value as sys(115).

Deployment under Vista

Due to the presence of the Program and Data directories in the Windows Vista installation of Omnis, you need to understand how the Omnis product tree is installed under Vista and therefore structure the installers for your own products a little differently.

Firstruninstall directory

For Windows Vista, the installed tree in Program Files must now have a directory called ‘firstruninstall’ in the Program directory. This contains the following directories and files, previously present in the Program directory:

Adhoc	Appbuild	Charmap.ini
Convert	Help	Html
Icons	Java	Local
Localclient	Startup	Studio
Welcome	Serial.txt (if present)	

After installation, when Omnis starts up for the first time, it copies the files in firstruninstall to the Data Directory – this happens in quite a short time, so no progress dialog is displayed while it occurs. Any files or directories in firstruninstall that are not present in the Data directory, are copied recursively to the Data directory.

Registry Settings

There is a new program called studiorg.exe. The installer runs this program in order to update the registry with the entries required by Omnis Studio beneath HKEY_LOCAL_MACHINE, such as file associations and event logging settings. Therefore, studiorg.exe must also be present in the Program directory, since Omnis can invoke it at startup if the registry needs updating.

When Omnis starts up on Windows Vista it first checks the registry to see if it is up to date (if for example you move the Studio tree, or run multiple copies of Studio from different locations, the registry needs changing to reflect the new path to omnis.exe). If the registry needs changing, Omnis invokes the program studiorg.exe. If Omnis is not running with administrative privileges (the default), it will display a message explaining that studiorg.exe needs to run, and after the user clicks OK, the Windows Vista UAC dialog will open, asking the user if studiorg.exe can run. Assuming the end user answers Yes, file associations and event logging are set up correctly.

System Fonts

Windows Vista introduces a new system font called ‘Segoe UI’ and uses a default font of Segoe UI 9 point. This can cause conflicts with the current default font on Windows XP which is Tahoma 8 point. The following changes have been made in Studio 4.3 to handle the new default font.

There is a new font name available in the Omnis Window & Report font tables, and the #STYLES system table, called ‘Omnis Windows System’ which represents the different system fonts under the different versions of Windows; therefore, on Windows XP it maps to Tahoma, and on Windows Vista it maps to Segoe UI. If you specify the Omnis Windows System font with a point size of *Zero*, Omnis will use 8 point Tahoma on Windows XP, and 9 point Segoe UI on Windows Vista.

The default Window and Report font tables on Windows now use Omnis Windows System for all new libraries. In addition, the default Omnis styles specified in #STYLES will use the new Omnis Windows System font.

Detecting Vista

There is a new function called *isvista()* which returns true when Omnis is running on Windows Vista; it returns false for all other platforms and earlier versions of Windows. It can be executed in the Web Client.

The following additions have been made to sys(6) and sys(7) to support Windows Vista although you should tend to use the new *isvista()* function: sys(6) returns N on Vista, while sys(7) returns 6.0.

Default button preference

The $osxflashdefaultbutton preference has been renamed to $flashdefaultbutton and controls whether or not default OS buttons animate on Windows Vista, as well as on Mac OS X. The current setting of this preference on Mac OS X is maintained.

Running different versions of Omnis under Vista

In the Windows Vista version of Omnis Studio 4.3 (or above), when *User Account Control* (UAC) is turned on, you get the following message when you start different versions of the Omnis executable, that is, either the Development or Runtime version of Omnis Studio:

"Omnis Studio needs to run studiorg.exe to install registry settings. User Account Control will ask you to allow studiorg.exe to run. Please allow it to run."

This problem cannot be overcome as there is only one set of registry entries associating file extensions (such as, lbs) with the executable; the UAC prompt is related to the executable associated with the file associations changing.

Instead, you can disable this file association checking and updating process by adding the entry "UpdateFileAssociations=0" to the General section of omnis.ini in the Data Directory under Windows Vista (the ..\AppData\Local\ folder), or the normal Omnis root folder on other Win32 platforms. You will need to create the omnis.ini file, if one does not exist, in the following format:

```
[General]
UpdateFileAssociations=0
```

This causes the file associations to be left set to the last executable that was allowed to update the registry, therefore stopping the UAC message on Windows Vista. This has the same effect on other Win32 platforms, but since there is no UAC and therefore no messages, the inclusion of the ini file entry may not be necessary.

Installing NTSERV under Vista

You will get an error message if you try to install NTSERV.EXE if you don't have Administrator Rights and User Account Control (UAC) is turned ON. Therefore, if you want to install NTSERV you need to have Administrator Rights (or turn UAC off). To do this with UAC turned On, you can Right-click the Command Prompt and select 'Run as Administrator' to install NTSERV without error.

Mac OS X

Omnis Studio 5.0, and above, fully supports Mac OS X 10.5.1 and 10.6 for Development and Runtime versions (note that 10.5 was codenamed “Leopard”, and 10.6 was codenamed “Snow Leopard” by Apple).

Detecting Leopard

There is a new function called *isleopard()* which returns true when Omnis is running on Mac OS X 10.5 (Leopard); it returns false for all other platforms and earlier versions of Mac OS X. It can be executed in the Web Client.

Linux

Omnis Studio 5.0, and above, fully supports development in Redhat 9.2, SuSE 9.2 or Ubuntu 6. Omnis Studio is likely to run on any Linux distributions that have version 2.0 or higher kernel, and xfree86 3.3.4 or higher.

Printing under Linux

Under the Linux version of Omnis Studio, you can print using a PDF printing device, called PrintPDF. The output of your Omnis reports are sent to a PDF file. PrintPDF can print Unicode characters.

The prerequisites for PrintPDF are:

- ❑ **Python**
  normally part of the Linux installation
- ❑ **Reportlab 2.0**
  installed as part of the Omnis RPM installation
- ❑ **PIL,** Python Imaging Library (version 1.1.4 or higher)
  You will need to download and install PIL from
  http://www.pythonware.com/products/pil

PrintPDF creates a folder called printpdf in the Omnis tree which it uses for temporary PNG files and Python scripts.

PrintPDF device parameters

Under Linux, the PrintPDF device is selected by default in the Omnis Print destination dialog, rather than the Printer destination. It has a number of parameters:

- ❑ **Output command (text)**
  See the automatic output parameter for details.

❑ **Automatic output (Boolean)**
If checked, the output command parameter must be completed. PrintPDF generates the PDF in a temporary file, and then spools the file using the command, after adding the temporary file pathname to it:

If the command does not include the token %f, PrintPDF appends a space, followed by the pathname of the temporary file to the command text.

If the command contains the token %f, then PrintPDF replaces all instances of %f with the temporary file pathname less the .pdf extension. This allows the command, for example, to look like this:

```
    /.../xpdf-3.01pl2-linux/pdftops -noshrink -nocenter -level3
%f.pdf;lp -d myprinter %f.ps;rm %f.ps
```

In this example, PrintPDF uses an alternative install of XPDF, to the default on the system - the command runs the pdftops filter, spools the PostScript, and then deletes the Postscript file. Note that PrintPDF deletes the temporary PDF file.

The parameters are:

%f = the pdf path (without the .pdf extension)
%d = printer name
%o = printer options

If automatic output is not checked, PrintPDF generates the PDF in the pathname identified by the Omnis preference $prefs.$reportfile. The pathname must identify a file with the .pdf extension.

❑ **Disable transparent background objects (Boolean)**
The various PDF to PS filters do not all handle transparency correctly.
You can disable transparency with this option. In practice, most reports can use this option checked. ext always overlays the background objects. XPDF 3.0.1 makes a much better job of transparency

❑ **Keep Python script and PNGs (Boolean)**
If checked, various temporary files are not deleted after generating the PDF. This may be useful for debugging and support purposes.

Printer management (CUPS)

Printing in the Linux version of Omnis Studio uses the Common UNIX Printing System (CUPS) to manage printer destinations, print jobs and their settings. According to the CUPS web site (www.cups.org): “CUPS provides a portable printing layer for UNIX based operating systems, … and is the standard printing system in Mac OS X and most Linux distributions. CUPS uses the Internet Printing Protocol (IPP) as the basis for managing print jobs and queues and adds network printer browsing and PostScript Printer Description (PPD) based printing options to support real-world printing.”

Python Imaging Library

The Python Imaging Library (PIL) does not need to be in the python sub-folder of the main Omnis folder. If you need python-imaging it must be installed with python, typically from the Linux operating system CDs.

Linux Fonts

Omnis uses the standard Fontconfig/Xft mechanism to manage fonts, which is supported by most other applications on Linux. This provides good support for scalable fonts, including anti-aliasing and support for Unicode character sets (although a Unicode version of Omnis for Linux is, at present, not available), both in the IDE and runtime environments. Omnis Studio no longer has a font cache.

As a result, the set of fonts available to Omnis under Linux is as follows:

Font	Has become
"-adobe-helvetica-"	"Luxi Sans" (if Luxi fonts are available) or "DejaVu Sans"
"-adobe-courier-"	"Luxi Mono" (if Luxi fonts are available) or "DejaVu Sans Mono"
"-adobe-times-"	"Luxi Serif" (if Luxi fonts are available) or "DejaVu Serif".

Font name conversion occurs for both the Linux font tables and font names used in $fontname property of the standard #STYLES system table. If a font is a TrueType font an icon is displayed against its name in the Change Font table dialog.

Note that the system substitutes a default font where the font name does not match one available, so text will be displayed even if its font does not exist.

Report fonts

The report fonts listed in #UXRFONTS comprise TrueType fonts only (those whose file has a .ttf extension). These are the most straightforward fonts to use with Reportlab, especially for Unicode.

Menu bar height/font

There are two properties in the setup section of OMNISXI.INI, which default as follows:

- ❑ **MenuBarHeight=22**
  MenuBarHeight is the height of the bar in pixels at 75dpi. It is scaled if the resolution is different.

- ❑ **MenuBarPointSize=11**
  MenuBarPointSize is the point size used for the menu bar font.

When you start up Omnis the defaults are written to OMNISXI.INI.

Font Location

The OMNISXIDIR environment variable can contain the path where the OMNISXI.INI and OMNISXI.FCI are stored. For example, you could set the variable using:

```
SET OMNISXIDIR=/tmp
```

Gnome & KDE environments

Omnis Studio supports the Gnome & KDE environments; no other environments are supported. To support native file dialogs, print dialogs, message boxes, and so on, a number of files are present in the root directory of Omnis Studio under Linux. There are two shared libraries, and an executable: omnisgnm.so, omniskde.so and omkdedlg. So when you deploy Omnis in the Runtime environment or using the Web Client, these files must be included. For example, the Web Client always uses the Gnome file omnisgnm.so, since the supported web browsers on Linux are GTK (Gnome)-based. All these files are supplied in the Studio development and Runtime environments by default.

File names

Under Linux, file names are treated as UTF-8 which means that path names containing accented characters are supported in the KDE and Gnome file managers.

Save window setup

The Save window setup option now saves and restores window positions correctly, although if you switch between Gnome and KDE you may notice that the main Omnis window is not the same height in both environments.

Events on inactive objects

Under the Linux version of Omnis, events do not pass through inactive objects on Linux as they do on Windows and Mac OS X. This is because in X11, mouse messages propagate through the chain of parent windows, rather than to the next window(s) in the Z order.

Other Platform Considerations

Setting Screen Resolution

The library preference $designdpi specifies the screen resolution in 'dots per inch' (DPI) of the current library, depending on the platform the library is running under. The preference stores a comma separated list of values in the following order: Windows DPI, Mac OS X DPI, Linux DPI, and the value for the current operating system is used as appropriate. If you assign a new value to this property, the new value does not take effect until you close and re-open the library.

This library preference allows for the fact that 96 DPI on Windows is equivalent to 72 on Mac OS X and 75 DPI on Linux; these are the standard screen resolutions (measured in DPI) at which Omnis Studio operates when creating or opening Omnis libraries.

Scaling occurs using the display DPI for the current platform, and the DPI value in $designdpi for the platform. Omnis will adapt the screen resolution of the library according to the current platform, that is, a library with design DPI set to 96, will display correctly to 96 on Windows, 72dpi on Mac OS X, and 75dpi on Linux.

One further effect of this preference is that the FontOps methods $wintextheight() and $wintextwidth() scale their result from the current display DPI, to $designdpi. This allows the results of these methods to be used correctly in conjunction with notation coordinates ($left, $top etc.), since these values all use the design DPI as their units.

The library preference $designdpimode specifies the behavior of the $designdpi preference. You assign a constant as follows:

- ❑ kDPIall
  Frames and fonts will be scaled
- ❑ kDPIframeOnly
  Frames will be scaled
- ❑ kDPIoff
  No scaling will be performed

If you assign this property, the new value does not take effect until you close and re-open the library.