Microsoft Licensing, Inc.
6100 Neil Road
Reno, NV 89511-1132

May 27, 1998

Microsoft **Licensing, Inc.**

Packard Bell/NEC
6041 Variel Avenue
Woodland Hills, CA 91367
USA
Attention: Mal Ransom

MICROSOFT CONFIDENTIAL

RE: MICROSOFT WINDOWS INITIAL BOOT PROCESS
Microsoft General OEM Business Terms Document effective November 1, 1997, between Microsoft Licensing, Inc. ("MS") and Packard Bell/NEC, ("COMPANY"), Microsoft license number 4861-5109340003 ("Business Terms Document")

Dear Mal Ransom:

The purpose of this letter and the attached documents is to describe an alternate process Packard Bell/NEC can employ during the Windows 98 initial boot process to complete the registration and ISP sign-up functions. Please note that if an alternate process is chosen for registration and/or ISP sign-up, the implementation and terms contained in the attached documents must be followed exactly. There are no additional alterations to the initial boot process offered or implied as a result of this communication.

There four documents in this package in addition to this cover letter are:
1) Summary letter providing details for the alternate registration and ISP sign-up process
2) Attachment #1 to the summary letter providing essential technical details
3) Attachment #2 to the summary letter -- the Authorized Registration Center (ARC) Kit
4) List of Registration Questions which is a supplement to the ARC Kit

If there are any questions or concerns about the optional alternate processes please contact the account manager assigned to your account.

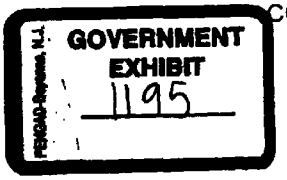
Sincerely,

Karen Hurlbut

Karen Hurlbut
General Manager OEM Operations
Microsoft Licensing, Inc.

cc: Thomas Henningsgard, Account Manager
Microsoft Corporation Law and Corporate Affairs
Microsoft Corporation License Administration
OEM Accounting Services Files

MS98 0113849
CONFIDENTIAL



RE: MICROSOFT WINDOWS 98 INITIAL BOOT PROCESS

Summary:

As part of Windows 98, the initial boot process guides the end user through valuable options so they can take full advantage of Windows 98. Joint product registration is one of the end user benefits which provides for a single registration process and is managed through MS authorized third parties. Additionally, the MS Internet Service Provider (ISP) Referral Server program provides OEMs the opportunity to use the MS ISP Referral Server as a means of offering quality ISPs to our mutual end user customers via a quick and easy sign-up process with a high rate of registration. Based on customer feedback received over the last several weeks on previous proposals, we have modified the registration and ISP sign-up portions of the initial boot process as a market test using a limited number of OEMs. Please read this entire document carefully to understand the guidelines to be followed in order to qualify for altering the initial boot process with COMPANY-specific processes. All defined terms shall have the meanings assigned to such terms in the applicable Microsoft Licensing, Inc. Agreement for Desktop Operating System Products (the "License Agreement").

Registration Details:

Notwithstanding the license limitations set forth in Additional Provision (18) of Exhibit C to the License Agreement, MS licenses COMPANY to implement an alternate OEM-supplied procedure for joint product registration during the Windows 98 initial boot process subject to the terms and conditions outlined in this document. In particular, and without limitation, this means that if COMPANY elects to utilize its own registration process, then:

- Any alternate COMPANY registration process must meet all specifications normally required from MS authorized third parties (Authorized Registration Center (ARC)) from time to time. Attachment 2 to this document (ARC Kit) provides a detailed description of these specifications: data to be captured, data formatting requirements, transmission specifications, and other particulars associated with obtaining all information necessary for a joint registration record to be complete. Attachment 2 applies to the current release of Windows 98; it may be updated at MS' discretion for future service releases of the product.
- COMPANY shall populate the system registry with the data that would otherwise have been written as a by-product of the standard MS registration process as incorporated into Windows 98. A complete list of valid system registry entries are documented in the ARC Kit.
- COMPANY's registration application shall run after the EULA, PID/COA entry screen, and Keyboard screens, and before the "Windows Welcome Screen" during the Windows 98 boot. This will precede the normal registration process and will execute as a Windows 98 RunOnce program. Alternatively, COMPANY may instead or in addition to the above, launch COMPANY's registration application on the Windows Welcome Screen by replacing the link between the Registration Button (Button 1) with a pointer to COMPANY's registration application. Notwithstanding the above, the MS Registration application code and support files must remain installed and unaltered on the Customer System hard disk.
- Any alternate registration process shall contain no third party advertising or product promotions other than those from COMPANY or COMPANY Subsidiary brands listed in the License Agreement.

- COMPANY's registration wizard must signify its successful completion (meaning the wizard executed in its entirety and came to a normal end of job) by removing the Register Now button on the Windows Welcome Screen in the same way as the standard joint registration process would alter the button upon completion. This will remind end users the registration process has already been completed when viewing the Windows Welcome Screen. Technical specifications explaining how to toggle attributes on the Windows Welcome Screen are contained in Attachment #1 to this document. If the COMPANY registration wizard does not come to a normal end of job for any reason, no modification to the appearance of the Windows Welcome Screen items can be made and the end user will have the option available to proceed with registration by selecting the Register Now button.

- Any registration wizard, application code or other intellectual property employed in the alternate registration process must be provided by COMPANY or its suppliers. MS' registration wizard code is reserved for use by end users utilizing the MS standard registration process.

Internet Service Provider Sign-up Details:

Notwithstanding the license limitations set forth in Additional Provision (18) of Exhibit C to the License Agreement, MS licenses COMPANY to implement an alternate OEM-supplied procedure for ISP sign-up during the Windows 98 initial boot process subject to the terms and conditions outlined in this document. In particular, and without limitation, this means that if COMPANY elects to utilize its own ISP sign-up process, then:

- COMPANY's ISP sign-up wizard application shall run after the EULA, PID/COA entry screen, and Keyboard screens, and before the "Windows Welcome Screen" during the Windows 98 boot. This will precede the normal ISP Referral Server process and will execute as a Windows 98 RunOnce program. COMPANY may combine the COMPANY's registration wizard and ISP sign-up wizard into a single program. In this case, COMPANY may activate that combined program as stated above or as a function of pressing the Register Now button on the Windows Welcome Screen. COMPANY may not replace the link under the Connect To The Internet button on the Windows Welcome Screen.

- COMPANY's ISP sign-up wizard must signify its successful completion (meaning the wizard executed in its entirety and came to a normal end of job) by toggling the Connect To The Internet button on the Windows Welcome Screen in the same way as the standard ISP sign-up process would check the button upon completion. This will remind end users selection of an Internet Service Provider has already been completed when viewing the Windows Welcome Screen. Technical specifications explaining how to toggle attributes on the Windows Welcome Screen are contained in Attachment #1 to this document. If the COMPANY ISP sign-up wizard does not come to a normal end of job for any reason, no modification to the appearance of the Windows Welcome Screen items can be made and the end user will have the option available to connect to the Internet utilizing the standard MS options provided through the welcome screen.

- COMPANY acknowledges that in the event COMPANY's ISP sign-up process fails to execute completely and/or come to a normal end of job, the end user customer will be presented with the standard Microsoft Welcome Screen which will invite the end user to Connect To The Internet.

- Upon failure per the above, and in order for COMPANY's ISP list to be displayed as a result of the end user selecting the Connect To The Internet button, the "custom option" must be activated. COMPANY acknowledges that the custom option requires a) separate contractual documents be executed with MS that provide for certain costs MS incurs to be recovered from COMPANY in return for maintaining COMPANY's custom ISP data on the MS referral server database, b) correct settings to be made

within the OEMINFO.INI file (further documented in the document titled "OEM's Guide to the Microsoft Internet Referral Service", and c) coordination of offers directly with the respective ISP to guarantee the integrity of the MS ISP Referral Server database.

- COMPANY shall disable the "Get Connected" desktop icon which invokes the MS Internet Connection Wizard, but only in the event COMPANY's ISP sign-up process has executed successfully and in its entirety (e.g. no fatal errors resulting from modem failure, absence of telecommunications connectivity, etc.). Instructions on how to disable this icon are contained in Attachment #1 to this document.
- Alternate ISP sign-up process shall contain no third party advertising or product promotions other than those from COMPANY, COMPANY Subsidiary brands, or the ISP being signed up.
- COMPANY shall leave the standard MS ISP sign-up application code (activated via the Windows Welcome Screen) and all supporting files installed and unaltered on the Customer System hard disk. Any COMPANY alternate ISP sign-up process must execute prior to the Windows Welcome Screen being displayed in the initial boot process (or as a continuation of the alternate registration process, however that alternate registration process is invoked per the options outlined in this document) and not a result of a redirected executable after selecting the Connect To The Internet option.
- Once both COMPANY's registration and ISP sign-up wizards signify their successful completion (meaning both wizards executed in its entirety and came to a normal end of job), the end user will land on the Windows Welcome Screen and the button labeled Discover Windows 98 will be the next logical step in the process.
- Any Internet connection wizard, application code or other intellectual property employed in the alternate registration process must be provided by COMPANY or its suppliers. MS' Internet connection wizard code is reserved for use by end users utilizing the MS standard registration process.

The decision to utilize an alternate first boot experience by following the guidelines in this document does not preclude COMPANY from creating a COMPANY-specific button on the bottom of the Windows Welcome Screen as provided for in Appendix A of the OPK Guidelines. In no event shall the purpose of any such COMPANY-specific button, if employed, provide for registration and/or Internet connectivity functions.

Attachment #1

Technical Details for the Product Registration Process

To remove the Register Now button from the Windows Welcome Screen
HKLM\Software\Microsoft\Windows\CurrentVersion\Welcome\Regwiz
@=1

To signify the completion of Product Registration
HKLM\Software\Microsoft\Windows\CurrentVersion
RegDone=1

Technical Details for the ISP Sign-up Process

To add the "check mark" on the Connect to the Internet Welcome button
HKLM\Software\Microsoft\Windows\CurrentVersion\Welcome\ICW
@=0x1 (DWORD)

To toggle the "Get Connected" desktop icon off and enable the the IE browser desktop icon
This cannot be done via the registry. Run ICWCONN1.EXE /restoredesktop

To signify the completion of ISP Signup
HKCU\Software\Microsoft\Internet Connection Wizard
Completed = 0x1(DWORD)

Microsoft Registration Wizard
OEM Support

Prepared by
AstraTek

March 16, 1998

1. Introduction

This document provides an overview of the processing performed by the Microsoft Registration Wizard when posting information to the Microsoft 'back-end' database.

The Registration Wizard (Regwiz) registers software products in a Microsoft database. It sends product information from the personal computer to Microsoft via:

- The Internet using a LAN connection
- The Internet using a modem connection

As part of its processing, Regwiz prompts information from the user and gathers information from the system about the hardware configuration.

As Regwiz receives information, it writes it to the registry. After it completes gathering the information, it reads the information from the registry and builds a data buffer containing the information to send to Microsoft.

Before Regwiz can send the information, it determines the mechanism it will use to transmit it, by checking to see if it can access a server at Microsoft via an Internet LAN connection. If that fails, it then checks to see if a modem is present. If a modem is present, it tries to contact the server at Microsoft via a dialup Internet connection. If it cannot access a modem, or if there is an error using the modem, the user is offered a choice to register by Yellow Card which is a printout of the information that can be sent to Microsoft via a mail carrier.

The Regwiz is supplied as an ActiveX control. The control has several properties that provide information that can be used by other utilities when sending information to other servers.

The information in this document covers:

- The Regwiz ActiveX method and properties
- The get system inventory information API GetSystemInventory
- The registry information written by Regwiz
- The steps used to send information to the Microsoft 'back-end' database

2. Regwiz ActiveX Properties and Method

The Regwiz is implemented as a Microsoft ActiveX component. Various properties and methods are available to provide an interface for applications to use.

2.1 Properties

The following properties can be used to provide information that is required when sending information to the 'back-end' database.

ActiveX binary name: REGWIZC.DLL

Version	Get	String	Returns the Registration Wizard version
MSID	Get	String	Returns the MSID
HWID	Get	String	Returns the HWID
IsRegistered	Get	BOOL	Returns TRUE if the product is registered or FALSE if it is not registered. NOTE: Before invoking, it is necessary to invoke the Put operation to specify the product that the registration status is required for.
	Put	String	Specify the product information in the registry. Note: for Windows 98 it is "SOFTWARE\Microsoft\Windows\Current Version"

2.2 Methods

Regwiz supports the following method.

InvokeRegWizard (STRING *ProductPath*) invokes registration for a particular product specified in the *ProductPath*.

The *ProductPath* string is of the following format: */n* ["*Registry entry for the product*"]

Where *n* can be the letter r or I .

- /r Re-register the product even if it has been registered.
- /I Register the product only if it's not registered.

Registry entry for the product is the string specifying the registry entry for the product. For example, for Windows 98, the string is:

SOFTWARE\Microsoft\Windows\Current Version

For the /r option, the *Registry entry for the product* can be null. If it is null, the operating system will be re-registered.

3. System Inventory Information

As part of the registration processing, Regwiz gathers information about the systems hardware configuration and stores it in the registry. The Regwiz uses the API GetSystemInventory, which is exported in the Microsoft provided dynamic link library SYSINV.DLL.

The API is defined:

BOOL WINAPI GetSystemInventory(INT *HwIndex*,
LPTSTR *szInventory*)

HwIndex is the index of the Hardware item
SzInventory is the string to receive the Hardware item description

The API returns TRUE if the hardware item is present and FALSE if it is not.

HwIndex can have the following values:

Defined constant	Value	Comment
INV_OEM	1	OEM Manufacturer Name
INV_PROCESSORTYPE	2	Processor Name
INV_TOTALMEMORY	3	Total RAM size
INV_TOTALHDSPACE	4	Total Hard disk space
INV_DISPRESOLUTION	5	Display Resolution
INV_DISPCOLORDEPTH	6	Display Color Depth
INV_WINVERSION	7	Operating System and its Version
INV_NETCARD	8	Network Adapter Card
INV_MODEM	9	Modem
INV_POINTDEVICE	10	Pointing Device
INV_CDROM	11	CD ROM
INV_SOUNDCARD	12	Sound Card
INV_REMOVEABLEMEDIA	13	Removable media
INV_COPRECESSOR	14	If Co-processor is present szInventory[0] = 1 for available 0 for non available

szInventory is a 255 byte string, allocated by the caller for the return value.

4. Post Data Format

After storing the information in the registry, the Regwiz sends the information to a Microsoft server via an HTML post command.

The format for the information, is URL encoded, key name/value pair syntax:

For example: KeyName1=Value&KeyName2=Value2...
Where:

- Key name/value pairs are separated from each other by an & character
- Key name fields are separated from the value fields by a = character
- All space characters in variable names or values are replaced by a + character
- Non alphanumeric characters are replaced by the % character followed by two hexadecimal digits representing the ASCII code of the non-alphanumeric character
- Line breaks are represented as either as carriage return (x0d) or carriage return/line feed (x0d/x0a)

The following table lists the key names sent by Regwiz.

Column definitions:

- Key Name is the name of the field being sent i.e. RegWizVer =
- Size is the length of the field, Var* indicates a variable length field
- Registry Key Name is the name in the registry of the key to retrieve the *value* of the key
- Description defines the field
- Further Processing describes any processing that is performed after the field is read from the registry and before it is sent to the back-end

All the registry keys are stored in:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\User information

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
RegWizVer	8	Version	Registration Wizard version Number	-
CodePage	5			-
LangCode	5	Language	Language code from GetSystemDefaultLangI D()	-
LangName	Var*	Language	Language Name	Use VerLanguageName( ) to convert the language code to actual language name
CreatedDate	10	No value, key name is sent blank	To be updated by the backend	-
RegDate	10	Date	Date of Registration	-
Fname	Var*	Default First Name	For far east countries, this is used for Name	-
Lname	Var*	Default Last Name	For far east countries this is used for Pronunciation	-
CompanyName	Var*	Default Company	Company name	-
AddrType	1	Default Company	Type of Address Home or Business	"1" - business or "2" – home If default company has a value, set to "1" otherwise set to "2"
Addr1	Var*	Mailing Address	Address information	-
Addr2	Var*	Additional Address	Address information	-
City	Var*	City	City	-
State	Var*	State	State	-
Zip	Var*	ZIP Code	Zip	-
CountryCode	4	Country	TAPI Country Id	The value returned from the Country key in the registry is the country index in the TAPI country information buffer. Use the index with CcntryInfo class method: GetTapiForTheCountryIndex to get the TAPI country Id.
Country	Var*	Country	Country Name	Use the class CcntryInfo method GetCountryName( ) to get the country name.
Phone		AreaCode+ Daytime Phone	Phone number. For FE countries it is Area Code and Phone number	Prefix Areacode with Daytime Phone and it should be separated by a – character. If Areacode is blank, then

KEY NAME	SIZE	REGISTRY KEY NAMES	DESCRIPTION	FURTHER PROCESSING
				Daytime Phone is the Phone number.
NoOther	1	Non-MS Products	Receive Non-Microsoft product service	-
Product	255	Application Name	Product being registered	-
PID	20	Product Identification	Product Identification	Strip out hyphens to produce 20 digit PID
OEM		OEM Manufacturer	OEM Manufacturer Name	-
SysInv	1	Include System	Send Hardware system inventory ? (Y/N)	-
OS	40	Operating System	Operating System	-
CPU	20	Processor	System Inventory info	-
MathCo	1	Math co-processor	System Inventory info	
TotalRAM	8	Total RAM	System Inventory info	Parse out the number
RAMUnits	2	Total RAM	System Inventory info	Parse out the unit
TotalDisk	8	Total Disk Space	System Inventory info	Parse out the number
DiskUnits	2	Total Disk Space	System Inventory info	Parse out the unit
RemoveableMedia	60	Removable Media	System Inventory info	-
DisplayRes	16	Display Resolution	System Inventory info	-
DisplayColorDepth	8	Display Color Depth	System Inventory info	-
PointingDevice	75	Pointing Device	System Inventory info	-
Network	75	Network	System Inventory info	-
Modem	75	Modem	System Inventory info	-
Sound	60	Sound Card	System Inventory info	-
CDROM	40	CD-ROM	System Inventory info	-
ProdInv	1	Include Products	Send Product Inventory (Y/N)	-
InvProd1	75	Product Inventory1	Product Inventory info	-
InvProd2	75	Product Inventory2	Product Inventory info	-
InvProd3	75	Product Inventory3	Product Inventory info	-
InvProd4	75	Product Inventory4	Product Inventory info	-
InvProd5	75	Product Inventory5	Product Inventory info	-
InvProd6	75	Product Inventory6	Product Inventory info	-
InvProd7	75	Product Inventory7	Product Inventory info	-
InvProd8	75	Product Inventory8	Product Inventory info	-
InvProd9	75	Product Inventory9	Product Inventory info	-
InvProd10	75	Product Inventory10	Product Inventory info	-
InvProd11	75	Product Inventory11	Product Inventory info	-
InvProd12	75	Product Inventory12	Product Inventory info	-
EmailName	50	E-mail Address	e-mail	-
Reseller	30	Reseller Name	Reseller name	-
ResellerCity	20	Reseller City	Reseller city	-
ResellerState	3	Reseller State	Reseller State	-
HWID	32	HWID	Hardware Id	-
MSID	32	MSID	MSID	-
Extension	32	Extension	Phone extension	-
DivisionName	50	DivisionName	Division name (far east countries only)	-
UserID	50	UserId	UserID is applicable for Japanese far east screen type.	-

4.1 Miscellaneous information

The Regwiz source file DCONV.CPP, contains the function:

PrepareRegWizTxbuffer(HINSTANCE hIns, char **tcTxBuf*, DWORD * *pRetLen*)

PrepareRegWizTxbuffer reads the registry information and constructs the URL encoded buffer to be posted to the back-end. This buffer can be posted directly to the back-end without further processing.

The buffer, *tcTxBuf* is allocated by the user. The buffer must be large enough to contain all the data. Currently, the buffer used in Regwiz is 4K bytes in length. The buffer length is specified in *pRetLen*. If the buffer is not large enough, the function returns the error, RWZ_BUFFER_SIZE_INSUFFICIENT, and *pRetLen* is set to the actual size required.

The Regwiz source file CNTRYINF.CPP, defines the class *CcntryInfo*. This class defines the methods *GetTapiIDForTheCountryInde*( ) and *GetCountryName*( ) which are referenced in the table above.

5. Overview of posting information to the Microsoft back-end

After Regwiz stores its information in the registry, it is sent to the Microsoft back-end, which is an Internet Information Server (IIS). The name of the server is: prodreg.microsoft.com

The information is sent via an HTTP post command. The Win32 Internet API is used to post the information.

The transmission of the Regwiz information to the back-end is done in five phases:

1) Determine how to connect to the back-end, prodreg.microsoft.com
2) Optionally, use a modem dialup connection to the Microsoft Network (MSN)
3) Prepare the data buffer for transmission
4) Use HTTP post via the Win32 Internet API
5) Update the registration status in the registry

5.1 Determine how to connect to the back-end

The Regwiz can send the information to the back-end via a dialup Internet connection or a Local Area Network (LAN) connection. Before the information can be sent, Regwiz determines which type of connection to use.

If a system is configured to use a LAN connection, Regwiz will try and use it to connect to the back-end. If it cannot access the back-end via the LAN connection, then it will check to see if a modem is present on the system. If a modem is present it will try and connect to the back-end via MSN.

When using the modem, Regwiz creates a temporary phone book entry and creates a dialup connection to the MSN.

If a modem is not present, then Regwiz invokes the appropriate setup programs for modem installation. If a modem is installed correctly, Regwiz will use it to connect to the MSN.

The Regwiz has been designed to handle tunneling through proxy servers that require authentication to be performed before the proxy server will allow the connection to continue. For proxy servers requiring authentication, the authentication is tried three times before the connection attempt is failed and the Regwiz attempts the connection via modem.

The Regwiz uses the function, *CheckInternetConnectivityExists* to determine how to connect to the back-end.

The function, *CheckInternetConnectivityExists*(HWND *hWnd*, HINSTANCE *hInstance*) is defined in the source file: RWPOST.CPP. This function returns three values for status:

1) CONNECTION_CANNOT_BE_ESTABLISHED - this value is returned if TCP/IP is not installed or a modem is not configured in the system

2) DIALUP_NOT_REQUIRED – is returned when a working LAN connection to the Microsoft registration site exists

3) DIALUP_REQUIRED – is returned when a connection to the Microsoft registration site could not be made via the LAN and a modem has been detected in the system

5.2 Using a modem for a dialup connection

When an Internet LAN connection is not available, Regwiz will check if a modem is configured on the system and use it to connect to the MSN.

The function *FdlgProc()* is defined in the source file UIDIAL.CPP and is the dialog procedure used to make a dialup connection. The dialog resource id is - IDD_DIAL. The Regwiz uses the Internet Connection Wizard (ICW) phone book table and finds the nearest MSN phone number for a particular country and area code and creates a temporary phone book entry. It makes a RAS connection using the phone book entry and checks for the connection status. Once a connection is established, it calls *SendHTTPData*() function defined in the source file, RWPOST.CPP. This function prepares the data buffer and posts it to the back-end.

5.3 Preparing a data buffer for transmission to the back-end

Once a connection to the back-end has been established, the next phase is to prepare the data buffer for transmission. The Regwiz function *PrepareRegWizTxbuffer* defined in DCONV.CPP, is called to prepare the data buffer for transmission.

The table in section four provides the details for the format and contents of the buffer formation.

5.4 Posting the data buffer to the back-end

The class *CinternetThread* defined in the Regwiz source file Ithread.CPP defines methods that simplify theWin32 Internet API. The sequence of methods used to send the data buffer is:

- SetBuffer(Buffer, BufferSize)
- SetSSLFlag(TRUE)
- PostData(hWnd)

Method definitions:

CinternetThread :: *SetBuffer* (LPSTR *pBuffer*, DWORD *dwLen*) - this method is used to specify the buffer that is sent in the HTTP POST command. *pBuffer* is the pointer to the starting address of the buffer. *dwLen* is the length of the buffer in bytes.

CinternetThread :: *SetSSLFlag(TRUE)* - this method is used to specify if SSL (Secure Sockets Layer) should be used in the posting of the information or not. If the posting has failed using SSL (*SetSSLFlag* set to TRUE), then the post operation is tried again without using SSL (*SetSSLFlag* set to FALSE).

CinternetThread :: *PostData(*HWND *hWnd)* - this method uses the buffer passed by *SetBuffer*() and uses the Win32 Internet API for the HTTP post.

6. Update the registration status in the registry

After Regwiz has successfully posted the information to the back-end, it will update the registry to indicate that the product registration has been completed. Regwiz does this by creating the registry key RegDone with a value of 1. This key is created in the product key of HKEYLOCALMACHINE. In Windows 98 the product key is:

HKLM\SOFTWARE\Microsoft\Windows\Current Version

7. Appendix A – Source code example

The following code 'snippet' illustrates the steps involved to post information to the back-end. All of the functions in the example can be found in the source (in the REGWIZPOST directory). The Regwiz uses a global instance of the class *CinternetThread* and the instance *theInternetClass* is defined as static in the source file, RWPOST.CPP.

The example code calls the functions, InitializeInetThread() and PostHttpData(). PostHttpData() checks the type of connectivity to the Internet, prepares the data buffer to be sent and posts it to the back-end.

Failures at any stage are returned as an error code. The Regwiz calls the function ExitWithXXXMessage() dialog in response to any errors that are returned.. These functions can be found in the source file RegWIzMain.cpp.

The following source code is a high level example of how Regwiz posts information to the back-end.

```
TransferRegWizInformation(HINSTANCE hInstance )
{
GetAutoDialConfiguration();        // If the connection is via Modem then  make sure not
                                   // to use the users ISP account, instead create  a temp
                                   // account with MSN. This function changes the
                                   // internet settings for AutoDial

    InitializeInetThread(hInstance);        // initializes theInternetClass

    DWORD dwRet = PostHTTPData(hInstance); // function  in rwpost.cpp

    // Based on the Return value display appropriate Error Message
    // These  Error Display Message functions are in RegWizMain.CPP

    switch (dwRet) {        // Has  Successfully send the info, update in Registry that  // the
                            product is registered
                            // Create RegDone key with "1" value in the corresponding //
                            Product Registry Key.
                            // e.g. for the Windows 98 Operating System  at
                            HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion
                            // creates a  Key  "RegDone" with a value of "1".

    case  RWZ_POST_SUCCESS:  // Update the registry with "RegDone"

    // Configuration Problem Hardware Setup Problem
    case  RWZ_ERROR_MODEM_NOT_FOUND :
    case  RWZ_ERROR_MODEM_CFG_ERROR :
    case  RWZ_ERROR_LOCATING_DUN_FILES :
          // Install or rectify and try later on
          ExitWithModemCfgError(hInstance,hParent);
          break;
```

```
    case RWZ_ERROR_NOTCPIP:
            ExitWithTcpCfgError(hInstance,hParent);
            break;

    // Configuration Problem or Installation Problem
    case RWZ_ERROR_LOCATING_MSN_FILES :
            ExitWithConfigurationProblem(hInstance,hParent);
            break;

    // Post Failure so try sending later on
    case RWZ_POST_FAILURE :
    case RWZ_POST_WITH_SSL_FAILURE :    // ?? 438 returned by MSN
    case RWZ_ERROR_NO_ANSWER :          // no answer
    case RWZ_POST_MSN_SITE_BUSY :
            ExitWithTryLater(hInstance,hParent);
            break;
            // Only Yellow Card as it is an internal Error
            // Usually this happens if there is a problem in making Data Buffer for
            // transmission

    case RWZ_NO_INFO_AVAILABLE:
    case RWZ_INVALID_INFORMATION:
    case RWZ_BUFFER_SIZE_INSUFFICIENT:
    case RWZ_INTERNAL_ERROR:
            ExitWithTxferError(hInstance,hParent);
            break;

    case RWZ_ERROR_INVALID_DLL:
    case RWZ_ERROR_INVALID_PARAMETER :
            ExitWithInputParamError(hInstance,hParent);
            break;


    case REGWIZ_ALREADY_CONFIGURED :
            ExitWithCompletedStatus(hInstance,szProductName);
            // Cancelled by user
            // User Has cancelled  in the Dialup connection screen

    case RWZ_ERROR_TXFER_CANCELLED_BY_USER:

    default :
            break;
    }

    ResetAutoDialConfiguration();       // This function restores the AutoDial
                                        // configuration

}
```

```
DWORD PostHTTPData(HINSTANCE hInstance)
{
        DWORD dwRet = RWZ_POST_FAILURE;
        DWORD dwRetStatus;

        dwRetStatus = CheckWithDisplayInternetConnectivityExists(hInstance,2);
                        // defined in InetDLG.CPP

        // This will display a Message Box window to display Checking for connection , //
        and calls CheckInternetConnectivityExists( ) and returns the value returned //  by it

        switch (dwRetStatus){
        case DIALUP_NOT_REQUIRED :
                // Posting via LAN
                //
                // PostDataWithWindowMessage function displays a MessageBox
                // window to display sending the information to the Registration site and
                // it calls
                // SendHTTPData(HWND hWnd, HINSTANCE hInstance) function   // and
                returns the return value of it

                if((dwRet = PostDataWithWindowMessage(hInstance))
                                == RWZ_POST_SUCCESS){
                        ;
                }
                else {
                        // Post Failure
                }
                break;

        case DIALUP_REQUIRED    :
                // Post Data via Dialup connection
                // Invoke the Dialog for Dialup connection defined in UDIAL.CPP
                // Once the Dialup makes a connection it calls SendHttpData()
                //
                dwRet=DialogBoxParam(hInstance, MAKEINTRESOURCE(IDD_DIAL),
                NULL,(DLGPROC)FDlgProc,
                        (LPARAM)hInstance);
                if(dwRet == -1 ) {
                        // Error in creating the Dialogue

                }

        case CONNECTION_CANNOT_BE_ESTABLISHED :
        default :
                // It is unexpected .

        break;
        }
        return dwRet;
}
```

MS98 0113867
CONFIDENTIAL

7.1 Source code program notes

The function PostHTTPData is defined in the source file, RWPOST.CPP. This function calls *CheckWithDisplayInternetConnectivityExists*() to verify the type of Internet connection to use. The function displays a message box advising that it is verifying the connection to the registration site. It then calls the function, *CheckInternetConnectivityExists*( ) which actually checks the connectivity.

If the return code DIALUP_NOT_REQUIRED is returned then the transmission is via the LAN. Regwiz calls the function *PostDataWithWindowMessage*() to display a message box advising the user that it's sending the information to Microsoft. It then calls the function *SendHTTPData*() to post the data buffer.

If the return code DIALUP_REQUIRED is returned, then the dialog procedure *FdlgProc*() is invoked to make a dialup connection. The *FdlgProc*( ) dialog procedure calls *SendHTTPData*( ) to post the data buffer to the back-end.

The function *SendHTTPData*() creates the data buffer and posts it to the back-end. It calls the *PrepareRegWizTxbuffer*( ) function for creating the data buffer. It then calls the *CInternetThread* class methods for posting the data buffer.

MS Registration Queries

OS Product Regwiz Schema and Profiling Questions

SLIDE 1: Welcome to RegWiz; copy regarding benefit of registration

SLIDE 2: Legal/privacy copy (see Exhibit B)

SLIDE 3: Product Owner

Specify the product owner, and then give your name and company name (if applicable).
I am registering this product for use at my:
__Home (primary use at home)
__Work (primary use at work)

First name: [variable text field]
Middle initial: [variable text field]
Last name: [variable text field]
Company name: Response option only if customer selects Work; if Home is selected this field is grayed out [variable text field]

SLIDE 4 a or b: Address and Phone Number

Depending on how the customer answers SLIDE 3 regarding where product is used (Home/Customer or Work/Company) determines the addressing question on SLIDE 4. It will also determine which taxonomy question (Work or Home) will populate the bottom section of SLIDE 5 (Taxonomy/Opt-out slide).

Slide 4a:
Company address: [variable text field]
Additional address information (e.g., apartment number): [variable text field]
City: [variable text field]
State or province: [variable text field]
Zip or postal code: [variable text field]
Area Code: [variable text field]
Phone number: [variable text field]
Email Address (e.g., johnsmith@microsoft.com): [variable text field]
Country: [drop down = ISO Standard]

or,
Slide 4b:
Home address: [variable text field]
Additional address information (e.g., apartment number): [variable text field]
City: [variable text field]
State or province: [variable text field
Zip or postal code: [variable text field]
Area Code: [variable text field]
Phone number: [variable text field]
Email Address (e.g., johnsmith@microsoft.com): [variable text field]
Country: [drop down = ISO Standard]

SLIDE 5 a or b: Taxonomy and Opt-out.

Which profiling question is decided by EU's response (Home or Work) on SLIDE 3.

SLIDE 5a:
Which category best describes your computing role in the organization where you work? (choose one only)
[Drop down menu response options]
Senior IT Decision-maker
IT Professional
IT Implementer/Analyst
Professional Developer
Web Developer□Other Developer
Senior Business Decision-maker
Line-of-Business Manager
Advanced Business Software User
General Business Software User□□
Occasionally, Microsoft offers its customers non-Microsoft products and services. Do you want to receive these offers? Yes/No [check box]

or,
SLIDE 5b:
Do your friends think of you as a knowledgeable source of information for computer software?
[Response option: Nine buttons; select one only] Rating scale of 1 – 9, with 9 = High Agreement

Are you excited about computers?
[Response option: Nine buttons; select one only] Rating scale of 1 – 9, with 9 = High Agreement

How would you rate your knowledge of personal computer software?
[Response option: Five buttons; select one only] Rating scale of 1 – 5, with 5 = High Knowledge

Occasionally, Microsoft offers its customers non-Microsoft products and services. Do you want to receive these offers? Yes/No [check box]

SLIDE 6: System Inventory result screen and EU approval to send
The Registration wizard found the following system information on your computer.
If you send Microsoft this system inventory, you will get more efficient product support.
System inventory: [max of 12]

Send system inventory with registration: Yes/No [check box]

SLIDE 7: End/Send; includes PID information.