

Windows Report Tool Deployment

Software Development Kit

Microsoft Corporation
Version 1.5
May 29, 1998

The Microsoft® Windows® Report Tool, Winrep.Exe, is an Internet-based problem reporting tool that is installed with Microsoft Windows 2000. This tool provides a means for uploading system information and a request for assistance over the Internet or an intranet to an OEM, corporate helpdesk, or support center.

This document outlines the necessary steps to implement a Windows Report Tool (WinRep) solution for corporate and OEM environments.

This document requires prior knowledge of the following concepts:

?1 Networking (TCP/IP)

?2 Internet Information Server (IIS)

For more information, visit the IIS site at:

[HTTP://Microsoft.Com/IIS](http://Microsoft.Com/IIS)

?3 Optional: Relational Databases (e.g.. Microsoft SQL Server)

For more information, visit the SQL site at:

[HTTP://Microsoft.Com/SQL](http://Microsoft.Com/SQL)

Contents

This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.

Microsoft Corporation may have patents or pending patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. The furnishing of this document does not give you any license to the patents, trademarks, copyrights, or other intellectual property rights except as expressly provided in any written license agreement from Microsoft Corporation.

Microsoft does not make any representation or warranty regarding specifications in this document or any product or item developed based on these specifications. Microsoft disclaims all express and implied warranties, including but not limited to the implied warranties of merchantability, fitness for a particular purpose and freedom from infringement. Without limiting the generality of the foregoing, Microsoft does not make any warranty of any kind that any item developed based on these specifications, or any portion of a specification, will not infringe any copyright, patent, trade secret or other intellectual property right of any person or entity in any country. It is your responsibility to seek licenses for such intellectual property rights where appropriate. Microsoft shall not be liable for any damages arising out of or in connection with the use of these specifications, including liability for lost profit, business interruption, or any other damages whatsoever. Some states do not allow the exclusion or limitation of liability or consequential or incidental damages; the above limitation may not apply to you.

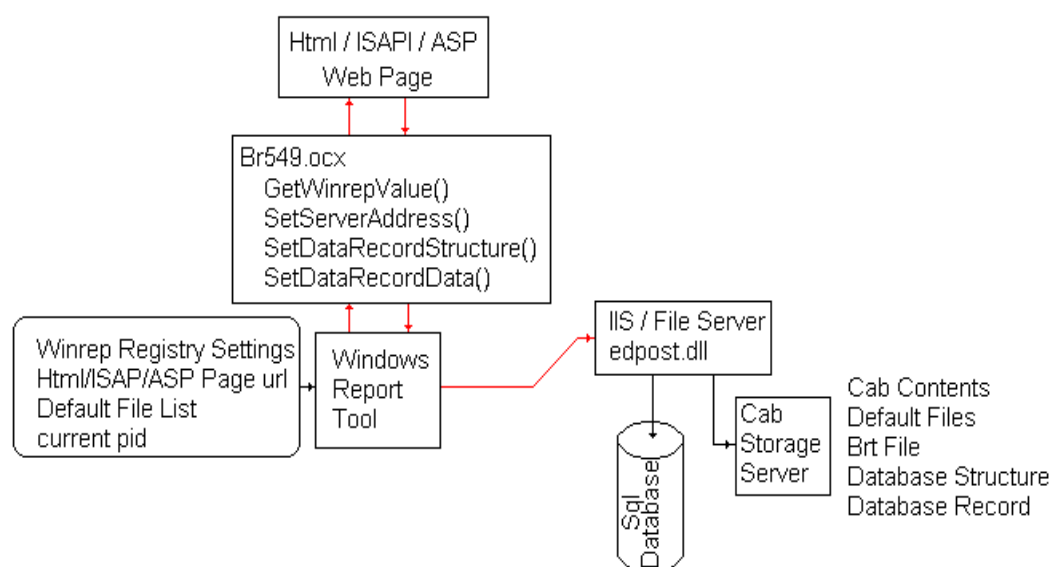
ActiveMovie, ActiveX, BackOffice, Developer Studio, Direct3D, DirectDraw, DirectInput, DirectPlay, DirectShow, DirectSound, DirectVideo, DirectX, Microsoft, NetMeeting, NetShow, Visual Basic, Win32, Windows, and Windows NT are trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries. Other product and company names mentioned herein may be the trademarks of their respective owners.

Overview of the WinRep Data Collection and

Uploading Process

WinRep is a tool that collects and uploads information detailing the computer's configuration to a server on the Internet or a company's Intranet. The figure below shows how WinRep can be programmed to gather and upload the information.

The WinRep Process for Uploading Problem Reports



- 1? When the user launches WinRep locally or from a Web page, data is collected from the client system, including a snapshot of the current configuration and the default file list from the registry. Also, WinRep may be programmed by a Web page with either the location of a second Web page or a path to a share on a file server, as well as a list of files to gather for the support incident.
- 2? WinRep displays an opening screen containing fields for the problem description, expected results, and steps to reproduce the problem. This screen also contains menu items for viewing and modifying the user's contact information and the file list to be included with the final report.
- 3? When the user clicks the **Next** button, WinRep either offers to save the report or, if programmed with a URL, displays a Web page that is specific to the organization. This page is hosted in an OCX control that provides Web browsing functionality and enables WinRep to communicate with the hosted page. The page can employ either JavaScript or VB script, and it provides WinRep with the location of the destination server and the ISAPI DLL backend.

This page can also create and pass a database insert query that conforms to the Structured Query Language (SQL) specification. This provides a means for

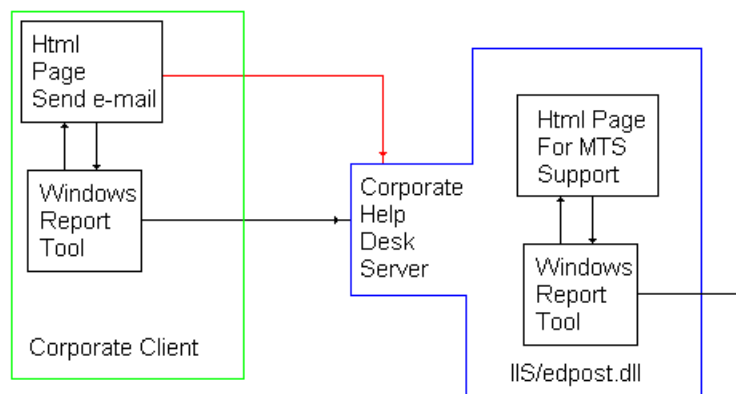
tailoring the database WinRep populates to meet your organization's requirements. If WinRep will not be populating a database, the second Web page can be used to send an e-mail message to the helpdesk, or to prompt the user for a filename that will identify the report to a Support Engineer.

- 4? Once the user has elected to send the report, WinRep closes the Web page and creates a compressed report as a cabinet (*.cab) file. This cabinet file contains copies of the files specified by the user or the WinRep calling command. All WinRep reports contain a Microsoft System Information file with a filename such as \$\$\$\$.nfo, that presents a comprehensive snapshot of the client machine's configuration at the time the report was generated. Microsoft System Information, MSInfo, can be used to open and examine the contents of both the WinRep cabinet file, and the MSInfo file it contains.
- ?4 The report is uploaded using an HTTP POST to an ISAPI DLL server extension that writes the report to a share on the Web server, and if applicable, writes a SQL query to a database.

Overview of the WinRep Support Solution

The process of implementing a WinRep solution for your support organization is made up of the following tasks:

- ?5 Configuring the Client Machine
 ?6 Creating the WinRep report page
 ?7 Setting up the Server



Corporate Help Desk Application of WinRep

The example above shows how WinRep might be implemented in a corporate environment. In this example, e-mail notifies the helpdesk of the request for service so no database is required. The helpdesk personnel open the report using Microsoft System Information, MSInfo, and examine it before contacting the user. If the company's helpdesk must escalate the issue to another support center, the same WinRep report can be attached to the escalation request.

Configuring the Client Machine

In this document, the *client* is the computer creating the WinRep report.

At run time, WinRep receives information from both the registry on the client computer and the command line arguments passed to it when WinRep is executed from a Web page.

?8 WinRep determines its default list of files to gather from the client's registry. The user may modify the list before saving or uploading the report.

?9 From a Web page, WinRep can be programmed with a URL for the location of the submission Web page, the Product's ID, a list of additional files to gather, or the location of an existing WinRep or DosRep report file (*.cab) to open and then upload.

Note

Command line parameters can be passed to WinRep directly from the command line, or they can be embedded in a Web page using the Br549.Ocx Run1 method. You should embed the WinRep calling command in a Web page using scripting—this allows for the most flexibility when developing a customized WinRep solution. If you choose to launch the tool locally, create a shortcut to WinRep.exe and add the command line parameters to it.

?10

Example: Launching WinRep from a Web page with command line parameters:

```
<HTML>
<TITLE>Test</TITLE>
<BODY>
<INPUT TYPE=button VALUE="Test Test()" ONCLICK="runMe()" NAME="Run Me">
<OBJECT
  CLASSID="clsid:167701E3-FDCF-11D0-A48E-006097C549FF"
  ID=test
  HEIGHT = 1
  WIDTH = 1
  HSPACE = 10
>
</OBJECT>
<SCRIPT LANGUAGE="JavaScript">

function runMe()
{
test.Run1(Error! Bookmark not defined."C:\\my file.txt\\");
}

</SCRIPT>
</BODY>
</HTML>
```

WinRep Command Line Parameters

WinRep has the following command line parameters:

```
Winrep.exe [target.htm] [report.cab] [/UNISQL] [/Categories: <category string>]
[/Files: <file1> [file2] [file3]]
```

The /FILES: switch must always be the last parameter passed. The following is an example of a command used to run WinRep with all of its parameters:

```
WinRep.exe //myserver/probrepo.htm C:/probrepo.cab /UNISQL /CATEGORIES:
+SystemSummar+SWEEnv /FILES:"%windrive%\Program Files\Accessories\Notepad.Exe"
"%windir%\win.ini"
```

Note

Microsoft System Information lists the Windows Report Tool on its Tools menu. This entry is maintained in the following registry key:

HKLM\Software\Microsoft\Shared Tools\MSInfo\Tools\WINREP

While it is possible to modify the local registry to program WinRep, a Web page should be used, as it offers the most flexibility. Also, configuring WinRep for a specific destination may lead to support requests for products that did not ship with the system.

Specifying the WinRep report page URL

WinRep determines that a command line string contains a URL by its extension. WinRep recognizes “.html,” “.htm,” or “.asp.” as valid extensions for a URL. This extension is not case sensitive. For example, if WinRep is passed the command line string, “http://www.myserver/mypage.htm” it will display the Web page at www.myserver/mypage.htm when the user presses the **Next** button.

If a URL path is not present when WinRep is executed, WinRep displays a Save As dialog when the **Next** button is clicked. This Save As dialog enables the user to save the report locally or to a file server on the LAN.

Specifying the initial CAB report file

At WinRep runtime, the command can also specify an existing report filename to open, that must end with a .CAB extension. WinRep will create a new report if no CAB file is specified.

Note

WinRep will only open and upload CAB files generated offline by WinRep or the MS-DOS Report Tool, DOSRep.exe.

Specifying a Unicode SQL query

Specifying /UNISQL on the command line will cause WinRep to issue a Unicode SQL query to the helpdesk SQL Server. SQL Server 7.0 SUPPORTS Unicode fields in a database. If /UNISQL is not specified, WinRep issues a standard non-Unicode SQL query. Edpost1.dll will support both types of SQL query.

Specifying categories to be included in NFO

Use the `/Categories:` command line switch to choose which categories are included in the report's NFO file. These rules apply to specifying categories:

- `" +category"` will add the category and its children to the set of output categories. Note, this also causes all the ancestors of the category to be added.
- `" -category"` will remove the category and its children from the set.
- `" +all"` will add all the categories to the set.

For example, `/Categories=+all-Components-SystemSummary` will generate an NFO with all categories except Components and System Summary. To obtain a list of categories, please see documentation for MSInfo.

Specifying files to be included with the report

The default list of files to upload with the report is drawn from both the client's registry and the command line. The registry settings for the default file list cannot be dynamically modified. The command line should be used to program WinRep with its default file list to allow for the greatest flexibility.

The command line can include the `/FILES:` switch to customize the default file list for the report. This switch is case sensitive and must be the last parameter passed to WinRep. All text on the command line following the `/FILES:` switch is treated as a series of files that are to be included with the WinRep report.

When using the `/FILES:` switch to select one of the files listed by default, the filename must be specified exactly as it appears in WinRep's "Files to copy:" list. For example, specifying `"/FILES: autoexec.bat"` will check the existing entry for the file in WinRep's default list. In contrast, specifying `"/FILES: Autoexec.bat"` would select nothing, and `"/FILES: C:\autoexec.bat"` would append an additional selected entry to the default list, leaving the original entry unchecked.

A file entry can begin with a UNC path (`\\server\share\`), a drive letter (`c:\`), or one of the following four environment variables:

Variable	Example	Actual Location
<code>%windir%</code>	<code>%windir%\myfile.txt</code>	The file is stored in the Windows directory of the computer
<code>%sysdir%</code>	<code>%sysdir%\myfile.txt</code>	The file, <code>myfile.txt</code> , is stored in the Windows System directory of the computer.
<code>%bootdir%</code>	<code>%bootdir%\myfile.txt</code>	The file, <code>myfile.txt</code> , is stored in the root directory of the computer's boot drive.
<code>%windrive%</code>	<code>"%windrive%\Program Files\myfile.txt"</code>	The file, <code>myfile.txt</code> , is stored in the Program Files directory on the drive containing Windows. (Note that long filenames must be enclosed within quotation marks).

Points to Remember

- ?11 Paths containing long filenames must be enclosed inside quotation marks (eg, "Long File Name" or \"Long File Name\" for Script).
- ?12 %windir%, %sysdir%, and %bootdir% variables are valid only at the beginning of the path string.
- ?13 WinRep has no variable for the Program Files directory. However files in this directory can be specified using the %windrive% variable (eg, \"%windrive%\\Program Files\\filename.ext\").
- ?14 If a file, such as autoexec.bat, exists in WinRep's default file list, specifying the filename on the command line selects the file for inclusion in the generated CAB if it is an exact match.
- ?15 The /FILES: switch is case sensitive.

Specifying File Size Limits for User Included Files

Two registry keys can be used to configure the maximum allowed size of the bug report and each file the user includes.

HKCU\Software\Microsoft\Winrep\Base\MaxCabSize - Defaults to 30MB

HKCU\Software\Microsoft\Winrep\Base\MaxIncludedFileSize - Defaults to 1MB

Creating the WinRep Report Page

The WinRep Support Solution requires the creation of a Web page that programs the client with the destination for the report, as well as its format. This Web page can be created using HTML, DHTML, or ASP; this allows you to customize the WinRep report as required by your support organization. For example, a support center that prefers to use e-mail in place of a database might write the WinRep report page to submit the contents of the HTML form to an ASP or CGI script that sends e-mail to the support alias instead of writing to a database.

The WinRep report page must include the information required by WinRep to send the report, and could reference WinRep's collected data. This communication between WinRep and the Web page it hosts is accomplished using the ActiveX control, Br549.Ocx.

Step 1: Load the Br549.Ocx Control

WinRep requires the Br549.Ocx control—the WinRep report Web page must load it for the tool to function. The following code loads the control:

```
<OBJECT CLASSID="clsid:167701E3-FDCF-11D0-A48E-006097C549FF"
  ID=Br549
  HEIGHT = 1
  WIDTH = 1
  HSPACE = 10>
</OBJECT>
```

Once Br549.Ocx is loaded, any information that WinRep collects can be referenced by the Web page. The data can be modified and included in the report WinRep uploads. This interaction is made possible by the Br549 methods

that can be called using either VB Script or JavaScript. Refer to Appendix A for the complete list of Br549.Ocx methods.

Step 2 (Optional) Create a SQL Query

The Web page can generate a service request using an insert SQL query to update a database on the server. This query is created using the `SetAddRecordQuery` method. The insert query should contain all of the data necessary to add a record to the server database. Once the query is constructed, it is passed back to WinRep with the `SetDBQuery` method. WinRep executes the query when the user chooses to send the report.

If the WinRep solution you implement does not require a database to manage the problem reports, the `SetDBQuery()` method is called with a blank string. If you choose to create a SQL query within the Web page, refer to the following example:

```
function SetAddRecordQuery()
{
    szQuery = "insert into table(description, name) Values(" +
    ""szDescription + "',' + "" + szName "" + ")";
    Br549.SetDBQuery(szQuery);
}
```

The Web page can call the Br549 method `GetWinrepValue` to obtain any of the data collected by WinRep. Refer to Appendix B for a list of all the values provided by WinRep.

For example: if the server database contains two fields, a description and a first name, then the Web page can reference these values with the following code:

```
<SCRIPT LANGUAGE="JavaScript">

szDescription = " ";
szName = " ";

function ItBegins()
{
    szDescription = Br549.GetWinrepValue("Description");
    szName = Br549.GetWinrepValue("Name");
}
...
</SCRIPT>
```

Step 3 Provide the destination URL and location of Edpost1.Dll

The Web page must program WinRep with the address of the destination server as well as the location of the ISAPI server extension, `Edpost1.Dll` (see the following section, "Setting up the Server," for more information concerning the ISAPI server extension). To do this, the Web page uses the Br549 `SetServerAddress` method illustrated below.

```
function SetDestination()
{
    Br549.SetServerAddress("http://www.myserver.com",
    "/isapi/support/edpost1.dll");
}
```

Note

A UNC path can be used in place of a URL as a parameter for `SetServerAddress` to designate a share on the LAN. In either case, this parameter must specify the location of the WinRep ISAPI server extension on a Windows NT IIS server.

Step 4 Create user controls for Sending and Canceling the report

Finally, the page uses the `Br549.Ocx ExitAndSend` function to instruct WinRep to begin uploading the report. Once the method is called, WinRep closes the Web page window and begins transmission of the report.

```
function SendReport()  
{  
    Br549.SendAndExit();  
}
```

As an alternative to the `SendAndExit` method, the Web page does not transmit the report if it calls the `Br549's Exit` method. This method closes the Web page window, but does not transmit the report. This is illustrated below:

```
function Cancel()  
{  
    Br549.Exit();  
}
```

See Appendix C for a sample Web page illustrating one implementation of steps 1-4.

Setting up the Server

In this document, the *server* is the computer receiving the WinRep report.

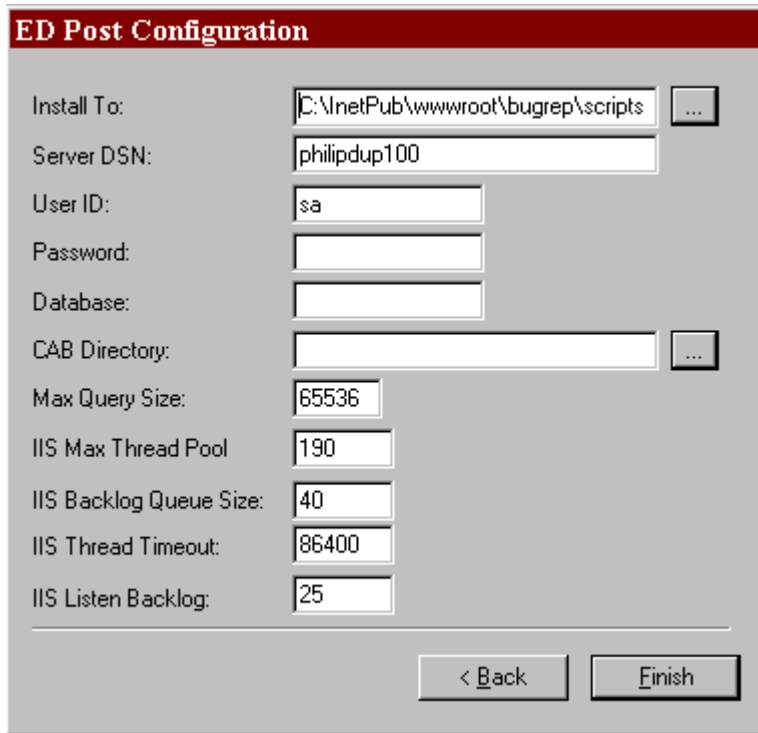
The WinRep client uses HTTP for both the file upload and optional SQL query. The ISAPI server extension, `EDPost1.dll`, is required to receive and write the cabinet file to a share on the server and, if applicable, write the SQL query to the specified database (DSN).

Step 1: Installing Epost1.Dll

WinRep requires the ISAPI server extension, `Edpost1.dll`, for all server-based solutions. `Edpost1.dll` is an Internet Service Application (ISA) DLL that is installed on the Windows NT IIS server by the ED Post Setup Wizard (`SetupEd.Exe`). This application can be found in the `<Cdroot>\Apps\Winrep` directory of the *Windows 2000 Resource Kit* compact disc.

`Edpost1.Dll` takes advantage of IIS' efficient thread management capabilities because it is an ISA. Therefore, a Windows NT server with IIS is a requirement of `SetupEd`.

When you run Setupd.Exe on the server, the following ED Post Configuration dialog box is displayed:



The image shows a dialog box titled "ED Post Configuration" with a red header bar. It contains several input fields and buttons. The fields are: "Install To:" with a text box containing "C:\inetPub\wwwroot\bugrep\scripts" and a browse button "..."; "Server DSN:" with a text box containing "philipdup100"; "User ID:" with a text box containing "sa"; "Password:" with an empty text box; "Database:" with an empty text box; "CAB Directory:" with an empty text box and a browse button "..."; "Max Query Size:" with a text box containing "65536"; "IIS Max Thread Pool" with a text box containing "190"; "IIS Backlog Queue Size:" with a text box containing "40"; "IIS Thread Timeout:" with a text box containing "86400"; and "IIS Listen Backlog:" with a text box containing "25". At the bottom, there are two buttons: "< Back" and "Finish".

Install To:	C:\inetPub\wwwroot\bugrep\scripts	...
Server DSN:	philipdup100	
User ID:	sa	
Password:		
Database:		
CAB Directory:		...
Max Query Size:	65536	
IIS Max Thread Pool	190	
IIS Backlog Queue Size:	40	
IIS Thread Timeout:	86400	
IIS Listen Backlog:	25	

< Back Finish

The information in the ED Post Configuration dialog box is used to install and configure Edpost1.Dll. In addition, the User ID, Password, and Database field are passed to Edpost1.Dll as parameters when it is first loaded. These fields should be maintained and updated when necessary using Setupd.Exe. The ED Post Configuration fields are as follows:

?16Edpost Configuration Field	?17Description
?18Install to	?19The server share where Edpost1.dll is to be installed.
?20Server DSN	?21The data source name (DSN) of SQL server database (if any).
?22User ID	?23The identification that the Edpost1.Dll uses to login to the database. For a SQL server administrator, this would be "sa."
?24Password	?25Used to specify the password Edpost1.Dll uses to login to the SQL database.
?26Database	?27The name of the SQL database to which the database records are stored.
?28CAB Directory	?29When this field is left blank, the WinRep client specifies the location of the cabinet file directory.
?30Max Query Size, IIS Max Thread Pool, IIS Backlog Queue Size, IIS Thread Timeout, and IIS Listen Backlog	?31These optional fields are used for performance tuning. For more information, see Appendix D.

Once the ED Post Configuration dialog box is filled in and the administrator clicks Finish, Setuped.Exe copies Edpost1.dll to the specified directory on the IIS computer.

Edpost1.dll can also be manually installed. This procedure requires that a registry file be created (see below). This registry file (*.reg) is merged into the registry on the IIS machine using the Regedit.exe application—the reg extension is associated with Regedit, so double-clicking the file will accomplish this.

Once the ED Post configuration information has been written to the registry, the Edpost1.dll must be manually copied to the IIS machine. This procedure allows a single server to securely support multiple database and cab stores—otherwise, this information is exposed on the WinRep report Web page.

```
REGEDIT4

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\EDPOST]
"Path"="E:\Inetpub\wwwroot"
"Server"=""
"Uid"=""
"Psw"=""
"Database"=""
"CabDirectory"=""
"MaxQuerySize"="8192"
```

Example registry file used to manually install Edpost.

Note

?32The IIS directory to which Edpost1.Dll is installed requires execute rights. Execute rights are granted though the Internet Server Manager Application. No other special rights are needed for Edpost1.Dll to function correctly. In fact, the FTP server need not be started or available for Edpost1.Dll to function.

?33The directory where the cabs are stored must have write access or Edpost1.Dll will not be able to create the uploaded reports.

?34

Step 2: (optional) Setting up the WinRep DSN

If the WinRep solution you choose to implement includes a database to manage service requests generated by WinRep, you must setup a System ODBC data source (DSN) configured for the WinRep database on the server.

An ODBC System data source stores information about how to connect to the indicated data provider. Any database that has an ODBC driver for Windows NT should be compatible with WinRep.

Appendix A

This section documents the Br549.Ocx API set. The Web page calls the APIs offered by the Br549.Ocx control to interact with WinRep.

The following table lists all the Br549.Ocx API methods.

Method	Description
AddUserFile	Enables the Web page to include a file in the .CAB file uploaded by WinRep.
CheckUserFile	Enables the Web page to check (include) or uncheck (exclude) a file from the WinRep-generated .CAB.
Exit	Informs WinRep that the Web page is finished and that the report should not be uploaded to the server.
GetTrackingNumber	Returns the tracking number that WinRep uses for the submitted report.
GetUserFile	Returns the full string path to an HTML user-added file.
GetUserFileIndex	Searches the user file array for a named file.
GetWinRepValue	Enables the Web page to read values directly from the WinRep application.
IsUserFileChecked	Searches the user file array for the named file.
Run1	Executes the Winrep.Exe client application with the specified command line. See text for a description of WinRep's command line processing
SendAndExit	Informs WinRep that the Web page is finished and that the report should be uploaded to the server.
SetBetaID	Sets an ID value that the Web page can use for user identification. The ID is stored on the client computer in an encrypted format.
SetDBQuery	Informs WinRep of the server database query to use to add a record to the server-side database.
SetRemoteCabName	Sets the name of the .CAB file that is to be created on the server.
SetServerAddress	Sets the location of the server to which the WinRep report is sent. This method also specifies the location of Edpost1.Dll.
SetTrackingNumber	Enables the WinRep-displayed Web page to designate a tracking number for the WinRep report.
Version	Returns the current version of Br549.Ocx.

AddUserFile

This method enables the Web page to add a file for inclusion in the .CAB file uploaded by WinRep.

The specified file is added only once. If the file has already been added, then this function simply returns. The file that is added is automatically checked for inclusion in the WinRep-created .CAB file. The Web page can use the CheckUserFile method to uncheck a user-added file. If the file is unchecked then WinRep does not include the file in the generated .CAB file.

If successful, this function returns the integer index of the newly added file; otherwise, it returns 0 because the array is full. There can be up to 255 user files stored in the user file array.

```
Integer
  AddUserFile(
    String szFileName
  )
```

Parameters:

szFileName

Filename of the file to be added.

CheckUserFile

This method enables the Web page to check (include) or uncheck (exclude) a file from the WinRep- generated .CAB.

A file must be checked in order to be included in the generated .CAB. The file is automatically checked when the AddUserFile method is used.

This method returns the previous checked state of the user file.

```
Boolean
  CheckUserFile(
    Integer index,
    Boolean bInclude
  )
```

Parameters:

index

Array index that identifies the user file to be included or excluded.

bInclude

TRUE if the file is to be included in the generated .CAB or FALSE if the file is to be excluded from the .CAB.

Exit

This method informs WinRep that the Web page is finished and that the report should not be uploaded to the server.

WinRep closes the HTML window when this method is invoked.

There is no return value.

```
Void
  Exit(
    Void
  )
```

Parameters:

None.

GetTrackingNumber

This method returns the tracking number that WinRep uses for the submitted report.

WinRep keeps track of submitted reports in its history list.

This method returns a string that contains the current tracking number of this report.

```
String
    GetTrackingNumber()
```

Parameters:

None.

GetUserFile

This method returns the full string path to an HTML user-added file.

If successful, this method returns the full path name of the file at the specified index; if the index is out of range or if there is not a file at the specified index, it returns an empty string.

```
String
    GetUserFile(
        Integer index
    )
```

Parameters:

index

The numeric index for which the Web page yields the file path name. This value can range from 1 to 255.

GetUserFileIndex

This method searches the user file array for the named file.

If successful, this method returns the index of the file in the user file array; otherwise, it returns 0 (the file is not found).

```
short
    GetUserFileIndex(
        String szFileName
    )
```

Parameters:

szFilename

The file to return the index for, expressed as a full path name.

GetWinRepValue

This method enables the Web page to read values directly from the WinRep application.

It returns the requested value if the key is one of the supported keys. Otherwise, it returns the string “Key szKey Unsupported\r\n”, where szKey is the name of the szKey parameter.

```
String
  GetWinrepValue (
    String szKey
  )
```

Parameters:

szKey

A string that specifies the values to be returned. For a list of supported values, see Appendix B.

IsUserFileChecked

This method searches the user file array for the named file.

This method returns TRUE if the file is checked; it returns FALSE if it is not checked or the file does not exist.

```
Boolean
  IsUserFileChecked(
    Integer index
  )
```

Parameters:

index

The array index of a file; the check state of this file is returned.

Run1

This method executes the Winrep.exe client application. The command line szCmdLine is passed to WinRep exe.

The WinRep application accepts a number of arguments on the command line:

?35The URL of the HTML Web page that WinRep displays when the Next button is pressed.

?36The name of a .CAB report file to load.

?37A list of files to include in the CAB. See “Setting up the Client Machine” in this paper for a complete description.

A string is returned. This string is blank if the execution of WinRep is successful. In the case of an error this string describes the error condition.

```
String
  Run1 (
    String szCmdLine
  )
```

Parameters:

szCmdLine

The command line passed to the WinRep client.

SendAndExit

This method informs WinRep that the Web page is finished and that the report should be uploaded to the server.

The **SetServerAddress** method must be called before this method ; otherwise, WinRep will not send the report. WinRep closes the HTML window when this method is invoked.

There is no return value.

```
Void
    SendAndExit(
        Void
    )
```

Parameters:

None.

SetBetaID

This method sets an ID value that the Web page can use for user identification. The ID is stored on the client computer in an encrypted format.

Returns TRUE.

```
Boolean
    SetBetaID(
        String szBetaId
    )
```

Parameters:

szBetaId

The new value used for the Beta ID.

SetDBQuery

This method informs WinRep of the server database query used to add a record to the server database.

Note: If the installation does not need the database functionality, this method should be called with a blank query string. There is no return value.

```
Void
    SetDBQuery(
        String szQuery
    )
```

Parameters:

szQuery

A string that represents the SQL query that adds a record to the server database.

SetRemoteCabName

This method sets the name of the .CAB file created on the server.

WinRep suggests a .CAB filename that is based on the client ID and microsecond counter. This ensures that the remote .CAB name is unique. If you use the HTML page to specify a name, the name must be unique to avoid collisions with other WinRep clients.

There is no return value.

```
Void
    SetRemoteCabName (
        String szCabName
    )
```

Parameters:

szCabName

The filename of the remote .CAB file.

SetServerAddress

This method sets the location of the server to which the WinRep report is sent. It also specifies the location of Edpost1.Dll.

There is no return value.

```
Void
SetServerAddress (
    String szServerUrl,
    String szEdpostLocation
)
```

Parameters:

szServerUrl

This string is set by the Web page to the address to which the report is to be sent. This string can be in UNC form if the server is on the local area network or an HTTP Internet address if the server is on the Internet.

szEdpostLocation

The location of the server dll, Edpost.dll.

SetTrackingNumber

This method enables the WinRep-displayed Web page to designate a tracking number for the WinRep report.

By default, WinRep assigns a unique 64 bit number to the tracking number field. To retrieve the current tracking number, use the **GetTrackingNumber** method.

There is no return value.

```
void
    SetTrackingNumber (
        String szTrackingNumber
    )
```

Parameters:

szTrackingNumber

String representation of the tracking number. This can be any alphanumeric string of up to 63 characters. .

Version

This method returns the current version of Br549.Ocx.

This method returns a string representing the version of Br549.Ocx.

```
String  
    Version()
```

Parameters:

None.

Appendix B

The values available through the GetWinrepValue method, and their key values, are listed in the following table.

Key Value	Returned Information
Name	Name of person submitting report
FirstName	First name of person submitting report
LastName	Last name of person submitting report
Email	E-mail name of person submitting report
Company	Company of person submitting report
Street	Street address of person submitting report
City	City of person submitting report
Zip	Zip code of person submitting report
State	State of person submitting report
Country	Country of person submitting report
DayPhone	Day or work phone number of person submitting report
NightPhone	Night or home phone number of person submitting report
FaxPhone	Fax phone number of person submitting report
OSVersion	OS version number in OS format
OSVersionNumber	OS version number, Raid compatible
OSProductId	PID of Windows
ProxyName	Proxy server name
ProxyPort	Proxy port address
CpuType	CPU code
Opsystem	Operating system
Compspeed	CPU speed
Memram	Amount of memory
ExplorerVersion	Version number of Explorer, if present
Drives	Disk drives
Bios	BIOS vendor
Compmake	Computer maker
Compmodel	Computer model
Display	Computer video cards make and model
Monitor	Computer monitor
Multimedia	Multimedia-specific info (sound)
Scsiadapter	SCSI adapters
Printers	Configured printers
Modem	Modem
Netcard	Net cards
Netclient	Net client configuration

Netservices	Net services installed
Mouse	Mouse
Keyboard	Keyboard
Pcmcia	PCMCIA card
Infrared	Infrared
Devicesusb	USB bus
Machineld	Machine ID
CDROM	CD-ROM info
Title	Report title
Description	Report description
CABFiles	Files included with .CAB
CabFileName	WinRep's suggested .CAB filename.
TrackingNumber	WinRep's suggested tracking number.
Language	The default installed OS language.
Area	Report area
BetaID	Beta ID

Appendix C

The following example Web page contains the functions necessary to be hosted by WinRep.

```
HTML>
<body bgcolor="white" text="#000000" link="#006633" vlink="#999900"
alink="#339966" onload="JMain()">
<!--mstheme--><font face="trebuchet ms, arial, helvetica">
<OBJECT
  CLASSID="clsid:167701E3-FDCF-11D0-A48E-006097C549FF"
  ID=Br549
  HEIGHT = 1
  WIDTH = 1
  HSPACE = 10
>
</OBJECT>

<STYLE>
H1 {font-family: "Tahoma"; font-size:32; text-align:left}
H2 {font-family: "Arial"; font-size:24; text-align:center}
H3 {font-family: "Arial"; font-size:20; text-align:right }
H5 {font-family: "Arial"; font-size:14; text-align:left; color:black }

.highlight { color:slateblue }
.shadow    { color:lightsteelblue }
.slabel    { color:black }

</STYLE>
<meta name="Microsoft Border" content="none">
</HEAD>

<TITLE>
  My Corporation's Submit Problem Report
</TITLE>
```

```
<DIV STYLE="position:absolute; top:13; left:18; width:400; height:100;"
>
<H1 CLASS="shadow">Windows 2000 Report</H1>
</DIV>
<DIV STYLE="position:absolute; top:10; left:15; width:400; height:100;">
<H1 CLASS="highlight">Windows 2000 Report</H1>
</DIV>
<BR>
<BR>

<FORM name="reportform">

  <DIV STYLE="position:absolute; top:55; left:15; width:45; " >
  <H5>Title:</H5>
  </DIV>

  <INPUT TYPE=TEXT ID=szTitle TABINDEX=1 TITLE="Descriptive problem
report title"
  STYLE="position:absolute; width=375; top=55; left=75" >

  <DIV STYLE="position:absolute; top:85; left:15; width:45; " >
  <H5>Name:</H5>
  </DIV>
  <INPUT TYPE=TEXT ID=szName TABINDEX=2 TITLE="Your name"
  STYLE="position:absolute; width=295; top=85; left=75" >

  <DIV STYLE="position:absolute; top:115; left:15; width:50; " >
  <H5>E-mail:</H5>
  </DIV>

  <INPUT TYPE=TEXT ID=szEmail TABINDEX=3 TITLE="Your e-mail address"
  STYLE="position:absolute; width=240; top=115; left=75" >

  <DIV STYLE="position:absolute; top:145; left:15; width:55; " >
  <H5>Beta ID:</H5>
  </DIV>
```



```

<INPUT TYPE=TEXT ID=szBetaID TABINDEX=4 TITLE="Your Beta ID"
  STYLE="position:absolute; width=80; top=145; left=75" >

<BUTTON ID=Send TABINDEX=5 TITLE="Send report to Microsoft"
  ONCLICK="SubmitIssue()";
  STYLE="position:absolute; width=55; top=180; left=75" >
  Send
</BUTTON>

<BUTTON ID=Cancel TABINDEX=5 TITLE="Cancel"
  ONCLICK="BailOut()";
  STYLE="position:absolute; width=55; top=180; left=135" >
  Cancel
</BUTTON>

</FORM>

<!--mstheme--></font>

<SCRIPT LANGUAGE="JavaScript">

function JMain()
{
  reportform.Send.focus();

  sz_betaid = Br549.GetWinrepValue("BetaId");

  sz_name = Br549.GetWinrepValue("Name");
  sz_trackingnumber = Br549.GetWinrepValue("MachineId");
  sz_title = Br549.GetWinrepValue("Title");
  sz_logfilename = Br549.GetWinrepValue("CabFileName");
  sz_logfilecontents = Br549.GetWinrepValue("CABFiles");
  sz_firstname = Br549.GetWinrepValue("FirstName");
  sz_lastname = Br549.GetWinrepValue("LastName");
  sz_company = Br549.GetWinrepValue("Company");
  sz_address1 = Br549.GetWinrepValue("Street");
  sz_address2 = "";
  sz_city = Br549.GetWinrepValue("City");
  sz_state = Br549.GetWinrepValue("State");
  sz_country = Br549.GetWinrepValue("Country");
  sz_zip = Br549.GetWinrepValue("Zip");
  sz_dayphone = Br549.GetWinrepValue("DayPhone");
  sz_nightphone = Br549.GetWinrepValue("NightPhone");
  sz_fax = Br549.GetWinrepValue("FaxPhone");
  sz_email = Br549.GetWinrepValue("Email");
  sz_product = "My Product";
  sz_version = Br549.GetWinrepValue("OSVersionNumber");
  sz_opersysver = sz_version;
  sz_issuetype = "1 - System Issue";
  sz_repro = "";
  sz_area = Br549.GetWinrepValue("Area");
  sz_subarea = "";
  sz_compat = "";
  sz_compatvndor = "";
}

```

```
sz_compatproduct = "";
sz_compatver = "";
sz_description = Br549.GetWinrepValue("Description");

sz_sample = Br549.GetWinrepValue("TrackingNumber");
sz_language = Br549.GetWinrepValue("Language");

sz_language = sz_language.substr(0, 20);

sz_opersystem = Br549.GetWinrepValue("Opersystem");

sz_compmake = Br549.GetWinrepValue("Compmake");
sz_compmodel = Br549.GetWinrepValue("Compmodel");
sz_memram = Br549.GetWinrepValue("Memram");
sz_cputype = Br549.GetWinrepValue("CpuType");
sz_compspeed = Br549.GetWinrepValue("Compspeed");
sz_drives = Br549.GetWinrepValue("Drives");
sz_printers = Br549.GetWinrepValue("Printers");

sz_bios = Br549.GetWinrepValue("Bios");
sz_display = Br549.GetWinrepValue("Display");
sz_monitor = Br549.GetWinrepValue("Monitor");
sz_multimedia = Br549.GetWinrepValue("Multimedia");
sz_cdrom = Br549.GetWinrepValue("CDROM");
sz_modem = Br549.GetWinrepValue("Modem");
sz_netcard = Br549.GetWinrepValue("Netcard");
sz_netclient = Br549.GetWinrepValue("Netclient");
sz_netservices = Br549.GetWinrepValue("Netservices");
sz_nettrans = Br549.GetWinrepValue("Nettrans");
sz_pcmcia = Br549.GetWinrepValue("Pcmcia");
sz_devicesusb = Br549.GetWinrepValue("Devicesusb");
sz_devices1394 = Br549.GetWinrepValue("Devices1394");
sz_infrared = Br549.GetWinrepValue("Infrared");
sz_mouse = Br549.GetWinrepValue("Mouse");
sz_keyboard = Br549.GetWinrepValue("Keyboard");
sz_scsiadapter = Br549.GetWinrepValue("Scsiadapter");
sz_idProgram = "254";
sz_otherhw = "";
sz_diskcomp = "";
sz_sysconfig = "";

reportform.szTitle.value = sz_title;
reportform.szName.value = sz_name;
reportform.szEmail.value = sz_email;
reportform.szBetaID.value = sz_betaid;
}

function SubmitIssue()
{
    if ( reportform.szTitle.value == "" )
    {
        alert("The report title needs to be filled in.");
        return;
    }

    if ( reportform.szName.value == "" )
```

```
{
    alert("The name field needs to be filled in.");
    return;
}

if ( reportform.szEmail.value == "" )
{
    alert("The Email field must be filled in.");
    return;
}

if ( reportform.szBetaID.value == "" )
{
    alert("The betaid field must be filled in.");
    return;
}

// ## Specify destination server and path to EDPOST1.DLL
Br549.SetServerAddress("myserver.com", "/isapi/support/edpost1.dll");

// ## Specify location and name for the report
cabName = Br549.GetRemoteCabName();
cabName = "/reportdir/" + cabName;
Br549.SetRemoteCabName(cabName);

sz_title = reportform.szTitle.value;

SplitName();

sz_email = reportform.szEmail.value;
sz_betaid = reportform.szBetaID.value;

Br549.SetBetaID(reportform.szBetaID.value);

// ## Specify database and table
szQuery = "use " + "DBname" + " insert into TableName(";

szQuery = szQuery + "betaid, ";
szQuery = szQuery + "trackingnumber, ";
szQuery = szQuery + "title, ";
szQuery = szQuery + "logfilefilename, ";
szQuery = szQuery + "logfilecontents, ";
szQuery = szQuery + "firstname, ";
szQuery = szQuery + "lastname, ";
szQuery = szQuery + "company, ";
szQuery = szQuery + "address1, ";
szQuery = szQuery + "city, ";
szQuery = szQuery + "state, ";
szQuery = szQuery + "country, ";
szQuery = szQuery + "zip, ";
szQuery = szQuery + "dayphone, ";
szQuery = szQuery + "nightphone, ";
szQuery = szQuery + "fax, ";
szQuery = szQuery + "email, ";
szQuery = szQuery + "product, ";
```

```

szQuery = szQuery + "version, ";
szQuery = szQuery + "issuetype, ";
szQuery = szQuery + "repro, ";
szQuery = szQuery + "area, ";
szQuery = szQuery + "subarea, ";
szQuery = szQuery + "compat, ";
szQuery = szQuery + "compatvendor, ";
szQuery = szQuery + "compatproduct, ";
szQuery = szQuery + "compatver, ";
szQuery = szQuery + "description, ";
szQuery = szQuery + "sample, ";
szQuery = szQuery + "language, ";
szQuery = szQuery + "opersystem, ";
szQuery = szQuery + "opersysver, ";
szQuery = szQuery + "compmake, ";
szQuery = szQuery + "compmodel, ";
szQuery = szQuery + "memram, ";
szQuery = szQuery + "cputype, ";
szQuery = szQuery + "compspeed, ";
szQuery = szQuery + "drives, ";
szQuery = szQuery + "printers, ";
szQuery = szQuery + "sysconfig, ";
szQuery = szQuery + "bios, ";
szQuery = szQuery + "display, ";
szQuery = szQuery + "monitor, ";
szQuery = szQuery + "multimedia, ";
szQuery = szQuery + "cdrom, ";
szQuery = szQuery + "modem, ";
szQuery = szQuery + "netcard, ";
szQuery = szQuery + "netclient, ";
szQuery = szQuery + "netservices, ";
szQuery = szQuery + "nettrans, ";
szQuery = szQuery + "pcmcia, ";
szQuery = szQuery + "devicesusb, ";
szQuery = szQuery + "devices1394, ";
szQuery = szQuery + "infrared, ";
szQuery = szQuery + "mouse, ";
szQuery = szQuery + "keyboard, ";
szQuery = szQuery + "diskcomp, ";
szQuery = szQuery + "scsiadapter, ";
szQuery = szQuery + "otherhw, ";
szQuery = szQuery + "idProgram) ";

szQuery = szQuery + " Values(";

szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_betaid) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_trackingnumber) +
"', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_title) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_logfilename) + "',
";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_logfilecontents) +
"', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_firstname) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_lastname) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_company) + "', ";

```

```

szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_address1) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_city) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_state) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_country) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_zip) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_dayphone) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_nightphone) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_fax) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_email) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_product) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_version) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_issuetype) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_repro) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_area) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_subarea) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_compat) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_compatvendor) + "',
";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_compatproduct) + "',
";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_compatver) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_description) + "',
";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_sample) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_language) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_opersystem) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_opersysver) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_compmake) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_compmodel) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_memram) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_cputype) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_compspeed) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_drives) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_printers) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_sysconfig) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_bios) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_display) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_monitor) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_multimedia) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_cdrom) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_modem) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_netcard) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_netclient) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_netservices) + "',
";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_nettrans) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_pcmcia) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_devicesusb) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_devices1394) + "',
";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_infrared) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_mouse) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_keyboard) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_diskcomp) + "', ";
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_scsiadapter) + "',
";

```

```
szQuery = szQuery + "'" + ConvertDoubleQuotes(sz_otherhw) + "', ";  
szQuery = szQuery + "254" + ")";  
  
Br549.SetDBQuery(szQuery);  
  
Br549.SendAndExit();  
}
```

```
function ConvertDoubleQuotes(szStr)
{
    szOut = "";

    for(i=0; i < szStr.length; i++)
    {
        szOut = szOut + szStr.charAt(i);
        if ( szStr.charAt(i) == "\"" )
            szOut = szOut + szStr.charAt(i);
    }
    return szOut;
}
function SplitName()
{
    sztmp = reportform.szName.value;
    for(i=0; i < sztmp.length; i++)
    {
        if ( sztmp.charAt(i) == " " )
        {
            sz_firstname = sztmp.substring(0, i);
            sz_lastname = sztmp.substring(i, sztmp.length);
            sz_firstname = sz_firstname.substring(0, 19);
            sz_lastname = sz_lastname.substring(0, 19);
            return;
        }
    }
    sz_firstname = sztmp;
    sz_lastname = "";
    sz_firstname = sz_firstname.substring(0, 19);
    sz_lastname = sz_lastname.substring(0, 19);
}
function BailOut()
{
    Br549.Exit();
}
</SCRIPT>
</BODY>
</HTML>
```

Appendix D

This section documents registry settings for the IIS and Web service.

A number of ISS and Web services may need to be adjusted for optimum performance when using the Edpost1.Dll Web Client. Using the Edpost1.Dll installation program (Setuped.Exe) is a convenient and safe method of adjusting the parameters. However, the parameters may also be changed manually in the registry. These parameters are described below.

WARNING

Using Registry Editor incorrectly can cause serious, system-wide problems that may require you to reinstall Windows to correct them. Microsoft cannot guarantee that any problems resulting from the use of Registry Editor can be solved. Use this tool at your own risk.

IIS Thread Pool Limit

The IIS thread pool contains all the pool threads currently in use by the ISA DLL. The following registry key:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet
  \Services\Inetinfo\Parameters\PoolThreadsLimit: (REG_DWORD)
```

Specifies the maximum number of pool threads that can be created in the system. Each pool thread watches for the network request and processes it. The default value for the maximum number of pool threads is 2 times the number of megabytes of memory on the system. For example, for a 128 MB system, this would amount to a maximum of 256 threads for IIS.

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet
  \Services\W3SVC\Parameters\AcceptExOutstanding: (REG_DWORD)
```

The IIS thread pool limit is a hard limit. When the maximum number of pool threads is reached, then no more threads will be available to handle incoming requests. At this point, the connection backlog will begin to grow. If the connection backlog fills up, ISS will start refusing incoming requests until the backlog is relieved. The default queue size is 40.

ThreadTimeout Interval

The following registry key:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet
  \Services\Inetinfo\Parameters\ThreadTimeout: (REG_DWORD)
```

Specifies the amount of time an input-output processing thread should be maintained even if there is no I/O activity on the system. In general, when there is no I/O activity and no requests are outstanding, the server is idle and does not consume memory. But if that situation exceeds the ThreadTimeout interval, then the thread is stopped. The ThreadTimeout interval is measured in units of

seconds, and the default is 24 hours. When using Edpost1.Dll, it is recommend that this parameter be decreased to allow idle and unnecessary IIS pool threads to be stopped.

ListenBacklog

The following registry key:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet
\Services\Inetinfo\Parameters\ListenBackLog: (REG_DWORD)
```

Specifies the maximum number of active connections to hold in the queue waiting for server attention. Enhanced Internet Information Server functionality generally makes it unnecessary to use or modify this key, although increasing this value to 50 can foster extremely heavy use . In the case of Edpost1.Dll, it may be necessary to increase this parameter to avoid connection problems when rapidly connecting large numbers of WinRep clients.