

# [MS-CER]: Corporate Error Reporting Version 1.0 Protocol Specification

---

## Intellectual Property Rights Notice for Protocol Documentation

- This protocol documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the protocols, and may distribute portions of it in your implementations of the protocols or your documentation as necessary to properly document the implementation. This permission also applies to any documents that are referenced in the protocol documentation.
- Microsoft does not claim any trade secret rights in this documentation.
- Microsoft has patents that may cover your implementations of the protocols. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. If you are interested in obtaining a patent license, please contact [protocol@microsoft.com](mailto:protocol@microsoft.com).
- The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.
- All other rights are reserved, and this notice does not grant any rights other than specifically described above, whether by implication, estoppel, or otherwise.

This protocol documentation is intended for use in conjunction with publicly available standard specifications, network programming art, and Microsoft Windows distributed systems concepts, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

A protocol specification does not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments you are free to take advantage of them.

## Revision Summary

Date	Revision History	Revision Class	Comments
05/11/2007	0.1		MCCP Milestone 4 Initial Availability
08/10/2007	0.2	Minor	Updated the technical content.
09/28/2007	0.3	Minor	Updated the technical content.
10/23/2007	0.4	Minor	Updated the technical content.

Date	Revision History	Revision Class	Comments
11/30/2007	0.5	Minor	Updated the technical content.
01/25/2008	0.5.1	Editorial	Revised and edited the technical content.

# Table of Contents

<b>1</b>	<b>Introduction .....</b>	<b>4</b>
1.1	Glossary .....	4
1.2	References .....	4
1.2.1	Normative References .....	4
1.2.2	Informative References.....	5
1.3	Protocol Overview (Synopsis).....	5
1.4	Relationship to Other Protocols.....	5
1.5	Prerequisites/Preconditions .....	5
1.6	Applicability Statement .....	6
1.7	Versioning and Capability Negotiation.....	6
1.8	Vendor-Extensible Fields .....	6
1.9	Standards Assignments.....	6
<b>2</b>	<b>Messages .....</b>	<b>7</b>
2.1	Transport .....	7
2.2	Message Syntax .....	7
2.2.1	Count.txt.....	7
2.2.2	Hits.log .....	7
2.2.3	CER File Share Folder Structure .....	8
2.2.4	Policy.txt .....	9
2.2.5	Status.txt .....	10
<b>3</b>	<b>Protocol Details .....</b>	<b>12</b>
3.1	Client to Server Detail.....	12
3.1.1	Abstract Data Model .....	12
3.1.2	Timers .....	12
3.1.3	Initialization .....	12
3.1.4	Higher-Layer Triggered Events.....	12
3.1.5	Message Processing Events and Sequencing Rules .....	12
3.1.6	Timer Events.....	12
3.1.7	Other Local Events .....	12
<b>4</b>	<b>Protocol Examples .....</b>	<b>14</b>
4.1	Application Fault Example.....	14
4.2	Kernel Fault Example .....	15
<b>5</b>	<b>Security .....</b>	<b>16</b>
5.1	Security Considerations for Implementers .....	16
5.2	Index of Security Parameters .....	16
<b>6</b>	<b>Appendix A: Windows Behavior .....</b>	<b>17</b>
<b>7</b>	<b>Index.....</b>	<b>18</b>

# 1 Introduction

This document specifies the Corporate Error Reporting Version 1.0 Protocol. This protocol is designed to enable businesses to manage all **error reporting** information within the organization. Through use of this protocol, problem reports generated on a set of client machines can be directed to a local or remote **CER file share** for analysis.

## 1.1 Glossary

The following terms are defined in [\[MS-GLOS\]](#):

**Universal Naming Convention (UNC)**  
**Augmented Backus-Naur Form (ABNF)**

The following terms are specific to this document:

**CER Client:** A client configured to use the Corporate Error Reporting Version 1.0 Protocol.

**CER File Share:** A designated folder that acts as the drop location for the **error reports** copied by the Corporate Error Reporting Version 1.0 Protocol.

**Error Report:** Information contained in a set of files that describes a problem event that has occurred on the system. The report is typically compressed into a single file for transmission.

**MAY, SHOULD, MUST, SHOULD NOT, MUST NOT:** These terms (in all caps) are used as described in [\[RFC2119\]](#). All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

## 1.2 References

### 1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact [dochelp@microsoft.com](mailto:dochelp@microsoft.com). We will assist you in finding the relevant information. Please check the archive site, <http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624>, as an additional source.

[LAVY-MEGGITT] Lavy, M. and Meggitt, A., "Windows Management Instrumentation (WMI)", Sams, 2001, ISBN: 1578702607.

If you have any trouble finding [LAVY-MEGGITT], please check [here](#).

[MS-SMB] Microsoft Corporation, "[Server Message Block \(SMB\) Protocol Specification](#)", July 2007.

[MS-WMI] Microsoft Corporation, "[Windows Management Instrumentation Remote Protocol Specification](#)", September 2007.

[RFC1035] Mockapetris, R., "Domain Names - Implementation and Specification", RFC 1035, November 1987, <http://www.ietf.org/rfc/rfc1035.txt>

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, <http://www.ietf.org/rfc/rfc2119.txt>

[RFC3986] Berners-Lee, T., Fielding, R., and Masinter, L., "Uniform Resource Identifier (URI): Generic Syntax", RFC 3986, January 2005, <http://www.ietf.org/rfc/rfc3986.txt>

[RFC4234] Crocker, D., Ed. and Overell, P., "Augmented BNF for Syntax Specifications: ABNF", RFC 4234, October 2005, <http://www.ietf.org/rfc/rfc4234.txt>

### 1.2.2 Informative References

[MSDN-CAB] Microsoft Corporation, "Microsoft Cabinet SDK", March 1997, <http://msdn2.microsoft.com/en-us/library/ms974336.aspx>

[MSFT-CER] Microsoft Corporation, "Corporate Error Reporting", <http://www.microsoft.com/licensing/sa/benefits/cer.msp>

## 1.3 Protocol Overview (Synopsis)

The Corporate Error Reporting Version 1.0 Protocol provides an organization with the ability to copy error reports from a set of client machines to a CER file share on a specified [Server Message Block \(SMB\) Protocol](#) file server with additional configuration options.

An error event, such as an application or kernel fault, causes the client system to collect information for an error report. The Corporate Error Reporting Version 1.0 Protocol does not create the original contents of the error report.

The **CER client** then performs a check to determine if a path to a CER file share has been specified for this client system. If a path to a CER file share has been specified, the CER client constructs a CER file share path based on the type of error that has occurred, and attempts to read specific configuration files on the CER file share that apply to this specific error. If any configuration files exist, the CER client reads the configuration information and adds the appropriate data to the error report. The CER client then compresses the report information into a single file. Finally, the CER client copies the error report to the specified CER file share, where it can be accessed by an administrator for tracking or analysis purposes.

The server is simply an SMB Protocol file server with a specific set of files. All behavior specific to the Corporate Error Reporting Version 1.0 Protocol is in the CER client.

## 1.4 Relationship to Other Protocols

The Corporate Error Reporting Version 1.0 Protocol uses the [SMB Protocol](#) to copy error reports from the client machine to the CER file share. There are no protocols that depend on the Corporate Error Reporting Version 1.0 Protocol.

## 1.5 Prerequisites/Preconditions

- The client system **MUST** to be able to create error reports.
- An implementation-specific file compression algorithm is required to organize the error reporting information into one file.
- The Corporate Error Reporting Version 1.0 Protocol assumes that the client is configured with the file path of the CER file share on the server.
- The Corporate Error Reporting Version 1.0 Protocol assumes that the client has permission to copy files to the CER file share.
- It is assumed that corporate administrators have the ability to interpret the files that the Corporate Error Reporting Version 1.0 Protocol copies to the CER file share on the [SMB Protocol](#) server.

## **1.6 Applicability Statement**

The Corporate Error Reporting Version 1.0 Protocol is not designed to be used by any other protocols. It is appropriate for small, medium, or large organizations that want to manage and review all error reporting information within the organization.

In addition, the Corporate Error Reporting Version 1.0 Protocol is only applicable to environments where all client machines support a common (local) file path syntax, although it is applicable to any such file path syntax.

## **1.7 Versioning and Capability Negotiation**

The Corporate Error Reporting Version 1.0 Protocol does not support any versioning.

## **1.8 Vendor-Extensible Fields**

The Corporate Error Reporting Version 1.0 Protocol does not support any vendor-extensible fields.

## **1.9 Standards Assignments**

The Corporate Error Reporting Version 1.0 Protocol does not use any standard assignments.

## 2 Messages

The following sections specify the message syntax for the Corporate Error Reporting Version 1.0 Protocol.

### 2.1 Transport

The Corporate Error Reporting Version 1.0 Protocol MUST use Server Message Block, as specified in [\[MS-SMB\]](#), to read and write files on the specified CER file share.

### 2.2 Message Syntax

The Corporate Error Reporting Version 1.0 Protocol transmits messages in the form of files.

#### 2.2.1 Count.txt

Count.txt includes two text parameters that track aggregate information about how frequently certain problems occur on specific clients and how much report data has been collected. It MUST conform to the following syntax, as specified in [\[RFC4234\]](#):

```
Countfile = Cabs Hits
Cabs      = "Cabs Gathered=" 1*DIGIT CRLF
Hits      = "Total Hits=" 1*DIGIT CRLF
```

**Cabs Gathered:** The number of error reporting files (these files can be in any format, the field name notwithstanding) that have been collected for this problem.

**Total Hits:** The total number of hits this problem has received.

#### 2.2.2 Hits.log

The hits.log file MUST conform to the following syntax, as specified in [\[RFC4234\]](#):

```
HitsLog = [HitEntry / HitEntry CRLF HitsLog]
HitEntry = Time HTAB Date HTAB Machine HTAB User HTAB FileName
Time     = Hours ":" Minutes ":" Seconds
Date     = Month ":" Day ":" Year
Hours    = 2DIGIT ; 00-23
Minutes  = 2DIGIT ; 00-59
Seconds  = 2DIGIT ; 00-59
Month    = 2DIGIT ; 01-12
Day      = 2DIGIT ; 01-28, 01-29, 01-30, 01-31 based on Month/Year
Year     = 4DIGIT
Machine  = 1*64(DIGIT / ALPHA) ;
FileName = 1*CHAR ;
```

**Time:** The local machine time when the report was generated (HH:MM:SS).

**Date:** The local machine date when the report was generated (MM-DD-YY).

**Machine:** The non-fully-qualified machine name of the machine that generated the error report.

**User:** The user name of the user who was logged on when the report was generated.

**FileName:** The name of the error reporting file that contains the report data. It MUST be in a file path notation supported by the client systems that are expected to encounter the type of error this file corresponds to. Each type of error has a separate hits.log file (specified in section [2.2.3](#)). The notation MUST support environment variables.

### 2.2.3 CER File Share Folder Structure

For user-mode error reports, the CER client MUST obtain the following information from the system to create this structure.

Parameter	Description
AppName	The file name of the faulting application binary.
AppVer	The file version of the faulting application binary obtained from the file descriptor.
ModName	The file name of the faulting module binary.
ModVer	The file version of the faulting module binary obtained from the file descriptor.
Offset	The line of code in the faulting binary where the exception occurred (in hexadecimal).

The terms below in brackets ("**<**" and "**>**") are not literals. Because the error report file can be of any type, the term "**<error reporting file>**" is substituted for the actual file name.

#### All Report Types

**<UNC file share path>\policy.txt**

#### Application Fault or Hang Reports

**<UNC file share path>\cabs\<AppName>\<AppVer>\<ModName>\<ModVer>\<Offset>\<error reporting file>**

**<UNC file share path>\cabs\<AppName>\<AppVer>\<ModName>\<ModVer>\<Offset>\hits.log**

**<UNC file share path>\status\<AppName>\<AppVer>\<ModName>\<ModVer>\<Offset>\status.txt**

**<UNC file share path>\counts\<AppName>\<AppVer>\<ModName>\<ModVer>\<Offset>\count.txt**

For other event types, the folder structure is dependent on the type of error event described in the report. "blue" MUST be used for kernel mode reports, and "shutdown" MUST be used for unplanned shutdown reports.

#### Kernel Mode Reports

**<UNC file share path>\cabs\blue\<error reporting file>**

**<UNC file share path>\cabs\blue\hits.log**

**<UNC file share path>\status\blue\status.txt**

**<UNC file share path>\counts\blue\count.txt**

#### Shutdown Reports

**<UNC file share path>\cabs\shutdown\<error reporting file>**



<UNC file share path>\cabs\shutdown\hits.log

<UNC file share path>\status\shutdown\status.txt

<UNC file share path>\counts\shutdown\count.txt

The Corporate Error Reporting Version 1.0 Protocol allows for a set of configuration parameters that describe which tracking or response options are preferred. These parameters are set in one of two text files located in specific locations on the CER file share, and instruct the CER client exactly what information to include in the error report. The configuration parameters MUST be contained in a [policy.txt](#) and [status.txt](#) file as specified in sections [2.2.4](#) and [2.2.5](#).

## 2.2.4 Policy.txt

If present, policy.txt MUST be placed in the root folder of the CER file share. It MUST conform to the following **Augmented Backus-Naur Form (ABNF)**, as specified in [\[RFC4234\]](#):

```
Policyrule           = [Tracking] [CrashesPerBucket] [URLLaunch]
                        [NoSecondLevelCollection] [NoFileCollection]
                        [NoExternalURL] [FileTreeRoot]
YesNoValue            = "YES" / "NO"
Tracking              = "Tracking=" YesNoValue CRLF
CrashesPerBucket      = "Crashes per bucket=" 1*DIGIT CRLF
URLLaunch             = "URLLaunch=" Url CRLF
Url                   = URI ; [RFC 3986]
NoSecondLevelCollection = "NoSecondLevelCollection=" YesNoValue CRLF
NoFileCollection      = "NoFileCollection=" YesNoValue CRLF
NoExternalURL         = "NoExternalURL=" YesNoValue CRLF
FileTreeRoot          = "FileTreeRoot=" Path CRLF
Path                  = 1*CHAR
```

**Tracking:** A value of "YES" MUST enable internal tracking in the form of a [hits.log \(section 2.2.2\)](#) file. A value of "NO" MUST disable internal tracking.

**Crashes per bucket:** The maximum number of error reporting files to collect for each error. If this many error reporting files already exist for this error on the CER file share, then an additional error reporting file MUST NOT be submitted.

**URLLaunch:** This parameter instructs the CER client that the specified URL contains more information about this problem. It MUST conform to the URL syntax, as specified in [\[RFC3986\]](#) Appendix A.

**NoSecondLevelCollection:** A value of "YES" instructs the CER client to ignore additional data requests specified in [status.txt \(section 2.2.5\)](#). A value of "NO" instructs the CER client to honor additional data requests.

**NoFileCollection:** A value of "YES" instructs the CER client to ignore file requests specified in status.txt. A value of "NO" instructs the CER client to honor file requests.

**NoExternalURL:** A value of "YES" instructs the CER client to ignore URL information described in status.txt. A value of "NO" instructs the CER client to honor URL information.

**FileTreeRoot:** This parameter instructs the CER client to put the report files in the specified location. It MUST be a **Universal Naming Convention (UNC)** path.

## 2.2.5 Status.txt

If present, status.txt MUST be placed in a final subfolder of the "status" branch of the CER file share (For an example, see section 4.1). The presence of a status.txt file in the final subfolder for a specific error instructs the CER client to take on different behavior for that error. The status.txt file supports any of the parameters that can be specified in [policy.txt](#), with the exception of FileTreeRoot. If there is a conflict, a status.txt parameter MUST override a corresponding policy.txt parameter.

The status.txt file supports all fields supported by policy.txt. It MUST conform to the following ABNF, as specified in [\[RFC4234\]](#):

```
StatusRule          = [Response] [RegKeyValues] [fDoc]
                      [WQLKeyValues] [GetFileKeyValues]
                      [GetFileVersionKeyValues][Tracking]
                      [CrashesPerBucket] [URLLaunch]
                      [NoSecondLevelCollection] [NoFileCollection]
                      [NoExternalURL] [FileTreeRoot]

Response            = "1" / Url CRLF
RegKeyValues        = "RegKey=" RegKeyList CRLF
RegKeyList          = (RegKey / RegKeyList ";")
RegKey              = 1*CHAR ;
fDoc                = "0" / "1" CRLF
WQLKeyValues        = "WQL=" WQLList CRLF
WQLList             = (WQL / WQLList ";")
WQL                 = 1*(ALPHA / "_" / DIGIT / WSP / DQUOTE / "*")
GetFileKeyValues    = "GetFile=" GetFileList CRLF
GetFileList         = (GetFile / GetFileList ";")
GetFile             = Path
GetFileVersionKeyValues = "GetFileVersion=" GetFileList CRLF
Path                = 1*CHAR
YesNoValue          = "YES" / "NO"
Tracking            = "Tracking=" YesNoValue CRLF
CrashesPerBucket    = "Crashes per bucket=" 1*DIGIT CRLF
URLLaunch           = "URLLaunch=" Url CRLF
Url                 = URI ; see note below
NoSecondLevelCollection = "NoSecondLevelCollection=" YesNoValue CRLF
NoFileCollection    = "NoFileCollection=" YesNoValue CRLF
NoExternalURL       = "NoExternalURL=" YesNoValue CRLF
FileTreeRoot        = "FileTreeRoot=" Path CRLF
Path                = 1*CHAR
```

**Note** The *Url* parameter uses URI Generic Syntax, as specified in [\[RFC3986\]](#).

**Response:** This parameter instructs the CER client to display a response prompt pointing to the URL specified by this parameter.

**RegKey:** This parameter lists any number of semicolon-delimited values to collect and include in the error report. [<1>](#)

**fDoc:** A value other than "0" instructs the CER client that the contents of any currently open word processing documents are requested to be added to the error report.

**WQL:** A string value that instructs the CER client to collect the Windows Management Instrumentation (WMI) objects (as specified in [LAVY-MEGGITT]) that are specified by this parameter, and include them in this error report.

**GetFile:** This parameter lists any number of semicolon-delimited file names to collect and include in the error report. It MUST be in a file path notation supported by the client systems that are expected to encounter the type of error this file corresponds to. Each type of error has a separate [hits.log](#) file (specified in section [2.2.3](#)). The notation MUST support environment variables.

**GetFileVersion:** This parameter lists any number of semicolon-delimited file names to collect version information from and include in the error report. It MUST be in a file path notation supported by the client systems that are expected to encounter the type of error this file corresponds to. Each type of error has a separate hits.log file (specified in section [2.2.3](#)). The notation MUST support environment variables.

**Tracking, Crashes Per Bucket, URLLaunch, NoSecondLevelCollection, NoFileCollection, NoExternalURL, FileTreeRoot:** These parameters are as specified in section [2.2.4](#).

## 3 Protocol Details

The following sections specify details of the Corporate Error Reporting Version 1.0 Protocol.

### 3.1 Client to Server Detail

#### 3.1.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

##### **DWFileTreeRoot**

The *DWFileTreeRoot* parameter specifies the UNC path to the location of the CER file share. The existence of this parameter instructs the CER client that the Corporate Error Reporting Version 1.0 Protocol will be used. [<2>](#)

#### 3.1.2 Timers

The Corporate Error Reporting Version 1.0 Protocol does not use any timers.

#### 3.1.3 Initialization

The CER client MUST check for the existence of the *DWFileTreeRoot* parameter. If there is no parameter, or if the parameter is invalid, the CER client MUST stop any further processing.

#### 3.1.4 Higher-Layer Triggered Events

The Corporate Error Reporting Version 1.0 Protocol does not use any higher-layer triggered events.

#### 3.1.5 Message Processing Events and Sequencing Rules

The Corporate Error Reporting Version 1.0 Protocol does not use any message processing events or sequencing rules.

#### 3.1.6 Timer Events

The Corporate Error Reporting Version 1.0 Protocol does not use any timers.

#### 3.1.7 Other Local Events

When a system or application error occurs, the CER client MUST perform the following actions:

- The CER client MUST first construct the file path depending on the type of error event, as specified in section [2.2.3](#). The CER client MUST then check for the existence of the relevant [policy.txt](#) and [status.txt](#) files, as specified in section [2.2.3](#), and if they exist then read the files by using the [SMB Protocol](#). It MUST verify that the files conform to the format specified in sections [2.2.4](#) and [2.2.5](#). If the files do not conform to the specified formats, the configuration options they specify MUST NOT be honored by the CER client.

- If both a status.txt file and a policy.txt file exist for a particular error and contain the same parameter with different values, the CER client MUST use status.txt parameter over the policy.txt parameter.
- If the [count.txt](#) file already exists, the CER client MUST attempt to increment the values in the existing count.txt file. If the policy.txt file indicated that internal tracking is enabled and [hits.log](#) already exists, the CER client MUST attempt to append the values to the existing hits.log file. If either of these attempts fail, the CER client MUST continue to copy the error report appropriately.
- If the status.txt file specifies additional data for the error report, the CER client MUST collect the requested data. The method of data collection is implementation-specific. [<3>](#) Note that there is no requirement that two clients use the same method or format of error information.
- The CER client MUST compress the complete report information into a single file by using any implementation-specific file compression. [<4>](#) Note that there is no requirement that two clients use the same file compression scheme.
- The CER client MUST attempt to copy the error report to the CER file share. If this attempt fails, the error report MUST NOT be copied.

## 4 Protocol Examples

The following topics describe problems that might occur on a client that is configured to upload to a CER file share.

### 4.1 Application Fault Example

1. An application fault occurs while running TestApplication.exe.
2. The system creates an error report.
3. The CER client checks to see whether a CER file share has been configured. The following value is set:

```
DWFileTreeRoot = "\\MyCerServer\CERFileShare\"
```

4. The CER client checks for the existence of a [policy.txt](#) file at the location specified by DWFileTreeRoot. No policy.txt file exists.
5. The CER client constructs the following folder structure based on the information specified in section [2.2.3](#):

```
\\MyCerServer\CERFileShare\status\TestApplication\1.0.0.0\TestModule\  
1.0.0.0\00000000\status.txt
```

6. A [status.txt](#) file exists at this location. The CER client parses the status.txt file, which includes the following parameters and values:

```
Tracking=YES  
Response=http://www.microsoft.com/ms.htm  
Crashes_per_bucket=100  
NoSecondLevelCollection=NO  
NoFileCollection=NO  
RegKey=HKLM\Software\Microsoft\PCHealth\ErrorReporting;HKLM\Software\  
Microsoft\PCHealth\Test  
fDoc=NO  
WQL=select * from Win32_logicaldisk  
GetFile=%WINDIR%\system32\notepad.exe;%WINDIR%\system32\faultrep.dll  
GetFileVersion=%WINDIR%\system32\notepad.exe;%WINDIR%\system32\  
faultrep.dll
```

7. This status.txt file has enabled internal tracking, so the CER client adds the following information to the [hits.log](#) file on the CER file share for this problem:

```
"15:32:23    04-23-07    TestMachine    TestUser    d5je031w.cab"
```

8. This status.txt file has specified a "Crashes per bucket" value of 100, so the CER client checks to make sure that 100 error reporting files have not already been collected for this problem.

9. This status.txt file has specified that additional data be added to the error report, in the form of two key values, a WMI query, two files, and version information for two files. The CER client collects this information and compresses all of the report files into a single file with the randomly generated name of "d5je031w".cab.

10. The CER client copies the following files to the CER file share:

```
\\MyCerServer\CERFileShare\cabs\TestApplication\1.0.0.0\TestModule\  
1.0.0.0\00000000\d5je031w.cab  
\\MyCerServer\CERFileShare\cabs\TestApplication\1.0.0.0\TestModule\  
1.0.0.0\00000000\hits.log  
\\MyCerServer\CERFileShare\counts\TestApplication\1.0.0.0\TestModule\  
1.0.0.0\00000000\count.txt
```

## 4.2 Kernel Fault Example

1. Kernel-mode fault occurs.
2. The system creates an error report.
3. The CER client checks to see whether a CER file share has been configured. The following value is set:

```
DWFileTreeRoot = \\MyCerServer\CERFileShare\
```

4. The CER client constructs the path for the [policy.txt](#) file:

```
\\MyCerServer\CERFileShare\policy.txt
```

5. The CER client attempts to read the policy.txt file, and finds that no policy.txt file exists.
6. The CER client constructs the path for the [status.txt](#) file. Because this is a kernel mode report, the path of status.txt is the following:

```
\\MyCerServer\CERFileShare\status\blue\status.txt
```

7. The CER client attempts to read the status.txt file, and finds that no status.txt file exists.
8. The CER client uses the [SMB Protocol](#) to copy the error reporting file to the specified file share. The following files are created on the CER file share:

```
\\MyCerServer\CERFileShare\cabs\blue\d5JE031w.cab  
\\MyCerServer\CERFileShare\counts\blue\count.txt
```

## 5 Security

The following sections specify security considerations for implementers of the Corporate Error Reporting Version 1.0 Protocol.

### 5.1 Security Considerations for Implementers

None.

### 5.2 Index of Security Parameters

The Corporate Error Reporting Version 1.0 Protocol does not use any security parameters.



## 6 Appendix A: Windows Behavior

The information in this specification is applicable to the following versions of Windows:

- Windows XP
- Windows Server 2003

Exceptions, if any, are noted below. Unless otherwise specified, any statement of optional behavior in this specification prescribed using the terms SHOULD or SHOULD NOT implies Windows behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that Windows does not follow the prescription.

For more information on how Windows implements the Corporate Error Reporting Version 1.0 Protocol, see [\[MSFT-CER\]](#).

[<1> Section 2.2.5:](#) Windows uses registry keys for these values.

[<2> Section 3.1.1:](#) On Windows XP and Windows Server 2003, the registry key

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\PCHealth\
ErrorReporting\DW] "DWFileTreeRoot"
```

specifies the UNC path of the CER file share.

[<3> Section 3.1.7:](#) Windows uses a new file for each piece of information collected (for example, .reg for a registry key or .mdmp for a minidump).

[<4> Section 3.1.7:](#) Windows uses .CAB files for this compression; for more information, see [\[MSDN-CAB\]](#).

## 7 Index

### A

[Abstract data model](#)  
[Applicability](#)  
[Application fault example](#)

### C

[Capability negotiation](#)  
[CER File Share fold structure](#)

### D

[Data model - abstract](#)

### E

Examples  
    [application fault example](#)  
    [kernel fault example](#)  
    [overview](#)

### F

[Fields - vendor-extensible](#)

### G

[Glossary](#)

### H

[Higher-layer triggered events](#)  
[Hits.log](#)

### I

[Implementer - security considerations](#)  
[Index of security parameters](#)  
[Informative references](#)  
[Initialization](#)  
[Introduction](#)

### K

[Kernel fault example](#)

### L

[Local events](#)

### M

[Message processing](#)  
Messages  
    [overview](#)  
    [syntax](#)  
    [transport](#)

### N

[Normative references](#)

### O

[Overview \(synopsis\)](#)

### P

[Parameters - security index](#)  
[Policy](#)  
[Preconditions](#)  
[Prerequisites](#)

### R

References  
    [informative](#)  
    [normative](#)  
    [overview](#)  
[Relationship to other protocols](#)

### S

Security  
    [implementer considerations](#)  
    [overview](#)  
    [parameter index](#)  
[Sequencing rules](#)  
[Standards assignments](#)  
[Status](#)  
[Syntax](#)

### T

[Timer events](#)  
[Timers](#)  
[Transport](#)  
[Tree management engine](#)  
[Triggered events - higher-layer](#)

### V

[Vendor-extensible fields](#)  
[Versioning](#)

### W

[Windows behavior](#)