# [MS-MERX]:

# Microsoft Error Reporting Extension to Corporate Error Reporting Version 1.0 Protocol

---

### Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation ("this documentation") for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.
- **Copyrights**. This 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 can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.
- **No Trade Secrets**. Microsoft does not claim any trade secret rights in this documentation.
- **Patents**. Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft Open Specifications Promise or the Microsoft Community Promise. If you would prefer a written license, or if the technologies described in this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **Trademarks**. The names of companies and products contained in this documentation might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names**. The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

**Reservation of Rights**. All other rights are reserved, and this notice does not grant any rights other than as specifically described above, whether by implication, estoppel, or otherwise.

**Tools**. The Open Specifications documentation 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. Certain Open Specifications documents are intended for use in conjunction with publicly available standards specifications and network programming art and, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

## Revision Summary

| Date | Revision History | Revision Class | Comments |
|---|---|---|---|
| 4/4/2008 | 0.1 | New | Initial Availability |
| 4/25/2008 | 0.2 | Editorial | Revised and edited the technical content |
| 6/27/2008 | 1.0 | Major | Revised and edited the technical content |
| 10/6/2008 | 1.01 | Editorial | Revised and edited the technical content |
| 12/12/2008 | 1.02 | Editorial | Revised and edited the technical content |
| 7/13/2009 | 1.03 | Major | Changes made for template compliance |
| 8/28/2009 | 1.04 | Editorial | Revised and edited the technical content |
| 11/6/2009 | 1.05 | Editorial | Revised and edited the technical content |
| 2/19/2010 | 2.0 | Editorial | Revised and edited the technical content |
| 3/31/2010 | 2.01 | Editorial | Revised and edited the technical content |
| 4/30/2010 | 2.02 | Editorial | Revised and edited the technical content |
| 6/7/2010 | 2.03 | Minor | Updated the technical content |
| 6/29/2010 | 2.04 | Editorial | Changed language and formatting in the technical content. |
| 7/23/2010 | 2.04 | None | No changes to the meaning, language, or formatting of the technical content. |
| 9/27/2010 | 2.04 | None | No changes to the meaning, language, or formatting of the technical content. |
| 11/15/2010 | 2.04 | None | No changes to the meaning, language, or formatting of the technical content. |
| 12/17/2010 | 2.04 | None | No changes to the meaning, language, or formatting of the technical content. |
| 3/18/2011 | 2.04 | None | No changes to the meaning, language, or formatting of the technical content. |
| 6/10/2011 | 2.04 | None | No changes to the meaning, language, or formatting of the technical content. |
| 1/20/2012 | 2.04 | None | No changes to the meaning, language, or formatting of the technical content. |
| 4/11/2012 | 2.04 | None | No changes to the meaning, language, or formatting of the technical content. |
| 7/16/2012 | 2.04 | None | No changes to the meaning, language, or formatting of the technical content. |
| 10/8/2012 | 2.04 | None | No changes to the meaning, language, or formatting of the technical content. |
| 2/11/2013 | 2.04 | None | No changes to the meaning, language, or formatting of the technical content. |
| 7/30/2013 | 2.04 | None | No changes to the meaning, language, or formatting of the |

| Date | Revision History | Revision Class | Comments |
|---|---|---|---|
| | | | technical content. |
| 11/18/2013 | 2.04 | None | No changes to the meaning, language, or formatting of the technical content. |
| 2/10/2014 | 2.04 | None | No changes to the meaning, language, or formatting of the technical content. |
| 4/30/2014 | 2.04 | None | No changes to the meaning, language, or formatting of the technical content. |
| 7/31/2014 | 2.04 | None | No changes to the meaning, language, or formatting of the technical content. |
| 10/30/2014 | 2.04 | None | No changes to the meaning, language, or formatting of the technical content. |
| 6/23/2016 | 2.04 | None | No changes to the meaning, language, or formatting of the technical content. |

# Table of Contents

# 1 Introduction

This document specifies the Microsoft Error Reporting Extension to Corporate Error Reporting Version 1.0 Protocol (MERX Protocol), a set of extensions to the Corporate Error Reporting Version 1.0 Protocol Specification, as specified in [MS-CER]. This specification assumes that the reader has familiarity with the concepts and requirements specified in [MS-CER]. Concepts and requirements specified in [MS-CER] are not repeated in this specification, except where required to specify how they are extended.

Sections 1.5, 1.8, 1.9, 2, and 3 of this specification are normative. All other sections and examples in this specification are informative.

## 1.1 Glossary

This document uses the following terms:

**ASCII**: The American Standard Code for Information Interchange (ASCII) is an 8-bit character-encoding scheme based on the English alphabet. ASCII codes represent text in computers, communications equipment, and other devices that work with text. ASCII refers to a single 8-bit ASCII character or an array of 8-bit ASCII characters with the high bit of each character set to zero.

**bucket**: A positive integer value that represents a mapping for a specific error signature.

**bucket table identifier**: A positive decimal integer that represents a mapping for a specific error signature.

**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.

**error signature**: An ordered collection of strings that represents an individual error or class of errors.

**error subpath**: A fragment of a directory path on a Server Message Block (SMB) Protocol file server that is composed of strings in an error signature and is used to direct error reports on the file share, as described in [MS-CER].

**globally unique identifier (GUID)**: A term used interchangeably with universally unique identifier (UUID) in Microsoft protocol technical documents (TDs). Interchanging the usage of these terms does not imply or require a specific algorithm or mechanism to generate the value. Specifically, the use of this term does not imply or require that the algorithms described in [RFC4122] or [C706] must be used for generating the **GUID**. See also universally unique identifier (UUID).

**Microsoft Error Reporting Extension (MERX) client**: A protocol client that is configured to use the Microsoft Error Reporting Extension to the Corporate Error Reporting Version 1.0 Protocol, as described in [MS-MERX].

**Microsoft Error Reporting Extension (MERX) file share**: A designated folder that stores error reports from the Microsoft Error Reporting Extension to the Corporate Error Reporting Version 1.0 Protocol, as described in [MS-MERX].

**registry**: A local system-defined database in which applications and system components store and retrieve configuration data. It is a hierarchical data store with lightly typed elements that are logically stored in tree format. Applications use the registry API to retrieve, modify, or delete registry data. The data stored in the registry varies according to the version of Windows.

**Unicode**: A character encoding standard developed by the Unicode Consortium that represents almost all of the written languages of the world. The **Unicode** standard [UNICODE5.0.0/2007]

provides three forms (UTF-8, UTF-16, and UTF-32) and seven schemes (UTF-8, UTF-16, UTF-16 BE, UTF-16 LE, UTF-32, UTF-32 LE, and UTF-32 BE).

**Uniform Resource Locator (URL)**: A string of characters in a standardized format that identifies a document or resource on the World Wide Web. The format is as specified in [RFC1738].

**Universal Naming Convention (UNC)**: A string format that specifies the location of a resource. For more information, see [MS-DTYP] section 2.2.57.

**UTF-16**: A standard for encoding Unicode characters, defined in the Unicode standard, in which the most commonly used characters are defined as double-byte characters. Unless specified otherwise, this term refers to the UTF-16 encoding form specified in [UNICODE5.0.0/2007] section 3.9.

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

## 1.2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the Errata.

### 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. We will assist you in finding the relevant information.

[DMTF-DSP004] Distributed Management Task Force, "Common Information Model (CIM) Infrastructure Specification", Version 2.3, October 2005, http://www.dmtf.org/standards/published_documents/DSP0004V2.3_final.pdf

[MS-CER] Microsoft Corporation, "Corporate Error Reporting Version 1.0 Protocol".

[MS-SMB] Microsoft Corporation, "Server Message Block (SMB) Protocol".

[MSDN-CAB] Microsoft Corporation, "Microsoft Cabinet Format", March 1997, http://msdn.microsoft.com/en-us/library/bb417343.aspx

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

[RFC5234] Crocker, D., Ed., and Overell, P., "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008, http://www.rfc-editor.org/rfc/rfc5234.txt

[UNICODE] The Unicode Consortium, "The Unicode Consortium Home Page", 2006, http://www.unicode.org/

### 1.2.2 Informative References

None.

## 1.3 Overview

This specification specifies a set of extensions to the Corporate Error Reporting Version 1.0 Protocol, as specified in [MS-CER]. These extensions add new capabilities to the Corporate Error Reporting Version 1.0 Protocol.

This specification uses the same section headings as [MS-CER] for straightforward interleaving of the base specification and the extension specification. The specific areas of extension are as follows:

- A minor extension to the Crash.Log file, specified in section 2.2.2.2 of this document.
- Many additional formats in the File Share Folder Structure section, specified in section 2.2.3 of this document.
- Several additional entries in the Status.Txt file, specified in section 2.2.5 of this document.
- An additional segment, section 2.2.6 of this document, which covers requirements for the error reporting file.
- Several minor changes to the Other Local Events segment, specified in section 3.1.7 of this document.
- Two additional example segments, found in section 4.3 and section 4.4 of this document, which reflect the extensions in this specification.

## 1.4 Relationship to Other Protocols

This protocol extends the original Corporate Error Reporting Version 1.0 Protocol to support additional kinds of error reporting, additional options for existing protocol details, and more specific requirements about error report contents.

There are no protocols that depend on the MERX Protocol.

## 1.5 Prerequisites/Preconditions

This section conforms to [MS-CER] section 1.5.

## 1.6 Applicability Statement

This section conforms to [MS-CER] section 1.6.

## 1.7 Versioning and Capability Negotiation

This section conforms to [MS-CER] section 1.7.

## 1.8 Vendor-Extensible Fields

This section conforms to [MS-CER] section 1.8.

## 1.9 Standards Assignments

This section conforms to [MS-CER] section 1.9.

# 2 Messages

The following sections specify the message syntax for the MERX Protocol.

## 2.1 Transport

The MERX Protocol MUST use the transport protocol specified in [MS-CER] section 2.1.

## 2.2 Common Message Syntax

The MERX Protocol transmits messages using the same method specified in [MS-CER] section 2.2. This section details the specific additions and changes to those messages for the MERX Protocol.

### 2.2.1 Count.Txt

The format of this file MUST be as specified in [MS-CER] section 2.2.1.

### 2.2.2 Tracking files

#### 2.2.2.1 Hits.Log

The format of this file MUST be as specified in [MS-CER] section 2.2.2.1.

#### 2.2.2.2 Crash.Log

The format of this file MUST be as specified in [MS-CER] section 2.2.2.2, with the following alteration to the "ErrorInfo" rule based on the **bucket**, the **bucket table identifier**, and **error subpath** values as specified in [RFC5234]:

ErrorInfo = BucketIDs / ErrorSubPath

BucketIDs = BucketID HTAB BucketTableID

BucketTableID = (%x31-39)*DIGIT / 0

**ErrorInfo:** The **MERX client** MUST write the **BucketIDs** to the Crash.Log file if it found a bucket in the Status.Txt file (section 2.2.5 of this document) for the error in question; otherwise it MUST write the error subpath for the error as specified in section 2.2.3 of this document.

**BucketTableID:** If the Status.Txt file (section 2.2.5 of this document) for the error in question contains a bucket table identifier, this MUST be that positive decimal integer. If the Status.txt file does not contain a bucket table identifier, this MUST be zero.

### 2.2.3 MERX File Share Folder Structure

This section conforms to [MS-CER] section 2.2.3, and specifies several additional formats supported by the MERX Protocol.

As in [MS-CER], the following terms in brackets ("<" and ">") are placeholders, not literals.

The MERX protocol supports two flexible error reporting models, simple error reporting (section 2.2.3.2.4 of this document) and generic error reporting (section 2.2.3.4 of this document). The parameters used in these types of reports MUST conform to the following syntax with **ASCII** characters, as specified in [RFC5234]:

Param = 1LeadingChar[1*254FollowingChar]

LeadingChar = (%d33-36) / (%d40-41) / (%d43) / (%d45-46) / (%d48-57) / (%d59) / (%d61) / (%d64-91) / (%d93-123) / (%d125-126) ; ASCII characters except " ", "%", "&", "'", "*", ",", "/", ":", "<", ">", "?", "\", "|", and "DEL".

FollowingChar = LeadingChar / %d32

**Param:** This string MUST conform to the requirements specified in [MS-CER] section 2.2 with respect to prohibited file names in addition to the specific characters called out in the ABNF, as specified in [RFC5234]<1>.

#### 2.2.3.1 Application Fault or Hang Reports

This type of report MUST conform to [MS-CER] protocol requirements, as specified in section 2.2.3.1. However, a MERX client SHOULD instead use the Extended Application Fault or Hang Report format, specified in section 2.2.3.3.

#### 2.2.3.2 Specialized Reporting Types

The MERX protocol supports several specialized reporting type formats in addition to those described in [MS-CER] section 2.2.3.2.

##### 2.2.3.2.1 Kernel Fault Reports

This type of report MUST conform to [MS-CER] protocol requirements, as specified in section 2.2.3.2.1.

##### 2.2.3.2.2 Shutdown Reports

This type of report MUST conform to [MS-CER] protocol requirements, as specified in section 2.2.3.2.2.

##### 2.2.3.2.3 Application Compatibility Reports

The error subpath for this type of report MUST be "appcompat", and the specific file **Universal Naming Convention (UNC)** paths used in making this type of report MUST be as follows.

- <UNC file share path>\cabs\appcompat\<error reporting file>
- <UNC file share path>\cabs\appcompat\Hits.Log
- <UNC file share path>\status\appcompat\Status.Txt
- <UNC file share path>\counts\appcompat\Count.Txt

##### 2.2.3.2.4 Simple Reports

To use the simple error reporting model, the error reporting software MUST specify a category name for the reports. The category name MUST conform to the "Param" rule specified in the introduction to section 2.2.3 of this document.

The error subpath for this type of report MUST be "simple\<category name>", and the specific file paths used in making this type of report MUST be as follows.

- <UNC file share path>\cabs\simple\<category name>\<error reporting file>
- <UNC file share path>\cabs\simple\<category name>\Hits.Log

- <UNC file share path>\status\simple\<category name>\Status.Txt
- <UNC file share path>\counts\simple\<category name>\Count.Txt

#### 2.2.3.2.5 Setup Error Reports

To use the setup error reporting model, the MERX client MUST obtain values for the following parameters from the software installation process<2>.

| Property | Description |
|---|---|
| ProdCode | This parameter represents the product code for the software installation. It SHOULD be a **GUID**. If the product code is not available, the value of this parameter SHOULD be the literal character "x" and MUST NOT be empty. |
| ProdVer | This parameter represents the product version for the software installation. If the product version is not available, the value of this parameter SHOULD be the literal character "x" and MUST NOT be empty. |
| Action | This parameter represents the name of action that caused the installation failure. If the action name is not available, the value of this parameter SHOULD be the literal character "x" and MUST NOT be empty. |
| ErrNum | This parameter represents the number of this particular error. If the error number is not available, the value of this parameter SHOULD be the literal character "x" and MUST NOT be empty. |
| Err0 | This parameter represents additional information about this error. If no further information is necessary, the value of this parameter SHOULD be the literal character "x" and MUST NOT be empty. |
| Err1 | This parameter represents additional information about this error. If no further information is necessary, the value of this parameter SHOULD be the literal character "x" and MUST NOT be empty. |
| Err2 | This parameter represents additional information about this error. If no further information is necessary, the value of this parameter SHOULD be the literal character "x" and MUST NOT be empty. |

The error subpath for this type of report MUST be

"setup\<ProdCode>\<ProdVer>\<Action>\<ErrNum>\<Err0>\<Err1>\<Err2>"

The specific file paths used in making this type of report MUST be as follows.

- <UNC file share path>\cabs\setup\<ProdCode>\<ProdVer>\<Action>\<ErrNum>\<Err0>\<Err1>\<Err2>\<error reporting file>
- <UNC file share path>\cabs\setup\<ProdCode>\<ProdVer>\<Action>\<ErrNum>\<Err0>\<Err1>\<Err2>\Hits.Log
- <UNC file share path>\status\setup\<ProdCode>\<ProdVer>\<Action>\<ErrNum>\<Err0>\<Err1>\<Err2>\Status.Txt

- <UNC file share path>\counts\setup\<ProdCode>\<ProdVer>\<Action>\<ErrNum>\<Err0>\<Err1>\<Err2>\Count.Txt

### 2.2.3.3 Extended Application Fault or Hang Reports

A MERX client SHOULD obtain additional, error-differentiating data for application fault or hang reports, and make an Extended Application Fault or Hang Report. In order for the MERX client to make such a report, it MUST obtain all of the following data in addition to that specified in [MS-CER] section 2.2.3.1:

| Property | Description |
|---|---|
| AppStamp | This parameter represents the timestamp of the build time of the faulting application binary. This MUST be represented as 8 hexadecimal digits without a leading 0x, for example, 1A2B3C4D. |
| ModStamp | This parameter represents the timestamp of the build time of the faulting module binary. This MUST be represented as 8 hexadecimal digits without a leading 0x, for example, 1A2B3C4D. |
| fDebug: | This parameter represents whether the executable includes detailed debugging information. The value "1" MUST be used to indicate True, while the value "0" MUST be used to indicate False. |

For this type of report, the error subpath MUST be

"<AppName>\<AppVer>\<AppStamp>\<ModName>\<ModVer>\<ModStamp>\<fDebug>\<Offset>"

The specific file paths used in making this type of report MUST be as follows.

- <UNC file share path>\cabs\<AppName>\<AppVer>\<AppStamp>\<ModName>\<ModVer>\<ModStamp>\<fDebug>\<Offset>\<error reporting file>
- <UNC file share path>\cabs\<AppName>\<AppVer>\<AppStamp>\<ModName>\<ModVer>\<ModStamp>\<fDebug>\<Offset>\Hits.Log
- <UNC file share path>\status\<AppName>\<AppVer>\<AppStamp>\<ModName>\<ModVer>\<ModStamp>\<fDebug>\<Offset>\Status.Txt
- <UNC file share path>\counts\<AppName>\<AppVer>\<AppStamp>\<ModName>\<ModVer>\<ModStamp>\<fDebug>\<Offset>\Count.Txt

### 2.2.3.4 Generic Error Reports

To use the Generic Error Reporting model, the error reporting software implementer MUST specify an EventTypeName and MUST specify an ordered set of parameters. There MUST be a minimum of one and a maximum of 10 parameters. The MERX client MUST combine the EventTypeName and parameters to create the error subpath directory fragment.

The EventTypeName and each parameter value MUST conform to the Param syntax, as specified in the introduction to section 2.2.3 of this document.

The 10 possible generic **error signatures** MUST be combined into error subpath strings as follows.

- 1 parameter: "generic\<EventTypeName>\<Parameter #1 Value>"
- 2 parameters: "generic\<EventTypeName>\<Parameter #1 Value>\<Parameter #2 Value>"
- 3 parameters: "generic\<EventTypeName>\<Parameter #1 Value>\<Parameter #2 Value>\<Parameter #3 Value>"
- 4 parameters: "generic\<EventTypeName>\<Parameter #1 Value>\<Parameter #2 Value>\<Parameter #3 Value>\<Parameter #4 Value>"
- 5 parameters: "generic\<EventTypeName>\<Parameter #1 Value>\<Parameter #2 Value>\<Parameter #3 Value>\<Parameter #4 Value>\<Parameter #5 Value>"
- 6 parameters: "generic\<EventTypeName>\<Parameter #1 Value>\<Parameter #2 Value>\<Parameter #3 Value>\<Parameter #4 Value>\<Parameter #5 Value>\<Parameter #6 Value>"
- 7 parameters: "generic\<EventTypeName>\<Parameter #1 Value>\<Parameter #2 Value>\<Parameter #3 Value>\<Parameter #4 Value>\<Parameter #5 Value>\<Parameter #6 Value>\<Parameter #7 Value>"
- 8 parameters: "generic\<EventTypeName>\<Parameter #1 Value>\<Parameter #2 Value>\<Parameter #3 Value>\<Parameter #4 Value>\<Parameter #5 Value>\<Parameter #6 Value>\<Parameter #7 Value>\<Parameter #8 Value>"
- 9 parameters: "generic\<EventTypeName>\<Parameter #1 Value>\<Parameter #2 Value>\<Parameter #3 Value>\<Parameter #4 Value>\<Parameter #5 Value>\<Parameter #6 Value>\<Parameter #7 Value>\<Parameter #8 Value>\<Parameter #9 Value>"
- 10 parameters: "generic\<EventTypeName>\<Parameter #1 Value>\<Parameter #2 Value>\<Parameter #3 Value>\<Parameter #4 Value>\<Parameter #5 Value>\<Parameter #6 Value>\<Parameter #7 Value>\<Parameter #8 Value>\<Parameter #9 Value>\<Parameter #10 Value>"

The specific file paths used in making this type of report MUST be as follows.

- <UNC file share path>\cabs\<ErrorSubPath>\<error reporting file>
- <UNC file share path>\cabs\<ErrorSubPath>\Hits.Log
- <UNC file share path>\status\<ErrorSubPath>\Status.Txt
- <UNC file share path>\counts\<ErrorSubPath>\Count.Txt

### 2.2.4 Policy.Txt

The format of this file MUST be as specified in [MS-CER] section 2.2.4.

### 2.2.5 Status.Txt

The use of Status.Txt is unchanged from that described in [MS-CER] section 2.2.5. The format of the file MUST be as specified in [MS-CER] section 2.2.5, with the changes and additions specified as follows, as specified in [RFC5234]. As in [MS-CER], note that the terms in the "StatusRule" rule can appear in any order and all permutations are not illustrated in the ABNF, as defined in [RFC5234], for brevity and clarity.

StatusRule = [Response] [DisplayType] [LightweightOptions]

[BucketID] [BucketTableID]

[iData] [RegKeyValues] [RegTreeValues] [fDoc]

[WQLKeyValues] [MemoryDump]

[GetFileKeyValues] [GetFileVersionKeyValues]

[Tracking] [CrashesPerBucket] [URLLaunch]

[NoSecondLevelCollection] [NoFileCollection]

[NoExternalURL]

; terms can appear in any order

DisplayType = %d68.105.115.112.108.97.121.84.121.112.101.61 ("0" / "1" / "2" / "3") CRLF ; the encoded characters spell case-sensitive "DisplayType="

LightweightOptions = %d84.114.105.100.101.110.116.79.112.116.105.111.110.115.61 LightweightOptionsValue CRLF ; the encoded characters spell case-sensitive "TridentOptions="

LightweightOptionsValue = 1*CHAR ; see the following for delimiter handling

BucketTableID = %d66.117.99.107.101.116.84.97.98.108.101.61 (%x31-39)*DIGIT CRLF ; the encoded characters spell case-sensitive "BucketTable="

RegTreeValues = %d82.101.103.84.114.101.101.61 RegKeyList CRLF ; the encoded characters spell case-sensitive "RegTree="

**DisplayType:** This term represents instructions on how to display the Response term. It is meaningful only given a MERX client that displays a user interface (UI).

- 0 informs the MERX client that it SHOULD show the Response **URL** as a link in its UI.
- 1 informs the MERX client that it SHOULD start the Response URL automatically.
- 2 informs the MERX client that it SHOULD indicate to the user that the Response URL is a user survey.
- 3 informs the MERX client that it SHOULD show the Response URL in a lightweight browser display UI.

This term MUST NOT be present if a Response term is not present in the Status.Txt file. If this term is not present and a Response term is present, the MERX client SHOULD use a default value of "0" for this term.

**LightweightOptions:** The text value associated with the LightweightOptions parameter represents configuration details for the lightweight browser display UI and is implementation dependent. This term MUST NOT be present if a DisplayType term with the value "3" is not present in the Status.Txt file.

**LightweightOptionsValue:** A CRLF pair MUST NOT appear in the LightweightOptionsValue, because that is reserved as the line delimiter.

**BucketTableID:** A positive decimal integer. This MUST be a positive integer, and MUST NOT be zero.

**RegTreeValues:** This parameter lists any number of semicolon-delimited registry key names. The MERX client MUST collect the values of these keys if they are present in the **registry**, as well as the values of all of their descendants. The MERX client MUST include this information in the error reporting file.

**GetFile:** In addition to the definition specified by [MS-CER] section 2.2.5, this file path notation MUST support wildcard characters. This means that an asterisk (*) in the file path can substitute for any zero or more characters, and the question mark (?) can substitute for any one character.

**GetFileVersion:** In addition to the definition specified by [MS-CER] section 2.2.5, this file path notation MUST support wildcard characters.

### 2.2.6 Error Reporting File

In the MERX Protocol, the name of the error reporting file MUST be generated to avoid naming conflicts within the "cabs\<ErrorSubPath>" directory (as specified in 2.2.3). The name of the error reporting file MUST end with ".Cab".

The format of the file SHOULD be the CAB file format as specified in [MSDN-CAB]. The MERX Client SHOULD include whichever types of data files are relevant for the error in question within the required single CAB file.

#### 2.2.6.1 Wql.Txt File

The MERX Client MUST generate and include a Wql.Txt file in the Error Reporting file if it finds a WQL parameter in the Status.Txt file (section 2.2.5 of this document) and if it can successfully execute the queries specified by the value of that parameter. This file MUST be a **Unicode** text file, as specified by [UNICODE].

The Wql.Txt file MUST conform to the following syntax, including **UTF-16** encoding, as specified in [RFC5234]:

WqlText = WqlItem / (WqlItem CRLF WqlText)

WqlItem = 1*CHAR ; UTF-16 encoding

**WqlItem:** the result of a single WQL query represented in Managed Object Format, as specified in [DMTF-DSP004]. All CR characters inside the resulting text MUST be transformed into CRLF pairs during construction of the Wql.Txt file.

# 3 Protocol Details

## 3.1 Client to Server Details

### 3.1.1 Abstract Data Model

This section is as specified in [MS-CER] section 3.1.1.

### 3.1.2 Timers

This section is as specified in [MS-CER] section 3.1.2.

### 3.1.3 Initialization

This section is as specified in [MS-CER] section 3.1.3.

### 3.1.4 Higher-Layer Triggered Events

This section is as specified in [MS-CER] section 3.1.4.

### 3.1.5 Message Processing Events and Sequencing Rules

This section is as specified in [MS-CER] section 3.1.5.

### 3.1.6 Timer Events

This section is as specified in [MS-CER] section 3.1.6.

### 3.1.7 Other Local Events

This section varies from that specified in [MS-CER] section 3.1.7.

When a system or application error occurs, if the client is configured to use the MERX Protocol (as specified in section 3.1.1 of this document), the client MUST perform the following actions, using this document as the reference for the format of each file:

1. This step MUST be as specified in [MS-CER] section 3.1.7, step 1.
2. This step MUST be as specified in [MS-CER] section 3.1.7, step 2.
3. This step MUST be as specified in [MS-CER] section 3.1.7, step 3.
4. This step MUST be as specified in [MS-CER] section 3.1.7, step 4.
5. This step MUST be as specified in [MS-CER] section 3.1.7 step 5, except that if the Status.Txt file additional data requests include a WQL parameter, the data gathered for that request MUST conform to the format specified in section 2.2.6.1 of this document.
6. This step MUST be as specified in [MS-CER] section 3.1.7 step 6, except the MERX client MUST compress the complete report information into a single error reporting file, which SHOULD be in the CAB file format (as described in section 2.2.6 of this document).
7. This step MUST be as specified in [MS-CER] section 3.1.7, step 7. In addition, the name of the copied error reporting file MUST end with ".Cab".
8. This step SHOULD be as specified in [MS-CER] section 3.1.7, step 8.
9. This step MUST be as specified in [MS-CER] section 3.1.7, step 9.

# 4 Protocol Examples

## 4.1 Application Fault

This example is as described in [MS-CER] section 4.1.

## 4.2 Kernel Fault

This example is as described in [MS-CER] section 4.2.

## 4.3 Extended Application Fault

1. An application fault occurs while running TestApplication.exe.
2. The system creates an **error report**.
3. The MERX client checks to see whether a **MERX file share** has been configured (as specified in section 3.1.1 of this document). The following value is set:

```
DWFileTreeRoot = "\\MyMERXServer\MERXFileShare\"
```

4. The MERX client checks for the existence of a Policy.Txt file at the location specified by DWFileTreeRoot. No Policy.Txt file exists.
5. The MERX client constructs the following folder structure based on the information specified in section 2.2.3.3 of this document:

```
\\MyCerServer\CERFileShare\status\TestApplication\1.0.0.0\00000000\TestModule\1.0.0.0\0000000
0\0\00000000\Status.Txt
```

6. A Status.Txt file exists at this location. The MERX client parses the Status.Txt file, which includes the following parameters and values:

```
Tracking=YES
Crashes per bucket=10
Bucket=12345
BucketTable=1
RegTree=HKLM\Software\Microsoft\PCHealth\ErrorReporting;HKLM\Software\Microsoft\PCHealth\Test
iData=1
GetFileVersion=%WINDIR%\system32\*.exe
```

7. This Status.Txt file has specified a "Crashes per bucket" value of 10, so the MERX client checks to make sure that 10 error reporting files have not already been collected for this problem. It does this by looking at the Count.Txt file for the error:

```
\\MyMERXServer\MERFileShare\counts\TestApplication\1.0.0.0\00000000\TestModule\1.0.0.0\000000
00\0\00000000\Count.Txt
```

The Count.Txt file has the following contents:

```
Cabs Gathered=6
Total Hits=17
```

Because 6 is fewer than 10, the MERX client continues the data collection process.

1. This Status.Txt file has specified that data should be collected for this error signature, and that additional data be added to the error report, specifically two registry tree enumerations and version information for all .Exe files in a particular directory. The MERX client collects this

information and compresses all of the report files into a single CAB-format file with the name of "Ov53rw8i.Cab".

2. The MERX client copies the error reporting file to the MERX file share:

```
\\MyCerServer\CERFileShare\cabs\TestApplication\1.0.0.0\00000000\TestModule\1.0.0.0\00000000\
0\00000000\Ov53rw8i.Cab
```

3. The MERX client updates the following file on the MERX file share to increment the number of hits and the number of copied error reporting files.

```
\\MyMERXServer\MERXFileShare\counts\TestApplication\1.0.0.0\00000000\TestModule\1.0.0.0\00000
000\0\00000000\Count.Txt
```

   The Count.Txt file now has the following contents:

```
Cabs Gathered=7
Total Hits=18
```

4. The Status.Txt file for this error signature has enabled internal tracking, so the MERX client opens the Crash.Log file on the MERX file share for this problem:

```
\\MyMERXServer\MERXFileShare\Crash.Log
```

5. The MERX client appends the following text to the Crash.Log file:

```
"15:32:23  04-23-2007   TestMachine   TestUser   12345   1"
```

6. The MERX client also opens the Hits.Log file on the MERX file share:

```
\\MyMERXServer\MERXFileShare\cabs\TestApplication\1.0.0.0\00000000\TestModule\1.0.0.0\0000000
0\0\00000000\Hits.Log
```

7. The MERX client adds the following information to the Hits.Log file on the MERX file share for this problem:

```
"15:32:23  04-23-2007   TestMachine   TestUser   Ov53rw8i.Cab"
```

## 4.4 Generic Error Reporting

1. An error occurs while installing TestProduct.
2. The TestProduct installer creates an error report. Its Generic EventTypeName is "TestProductSetup", and that EventTypeName is defined to have 3 parameters. The TestProduct installer determines that for this particular event, the first parameter has the value "0", the second has the value "1.0.0.0", and the third has the value "sample".
3. The MERX client checks to see whether a MERX file share has been configured as specified in section 3.1.1 of this document. The following value is set:

```
DWFileTreeRoot = "\\MyMERXServer\MERXFileShare\"
```

8. The MERX client checks for the existence of a Policy.Txt file at the location specified by DWFileTreeRoot. No Policy.Txt file exists.

9. The MERX client constructs the following folder structure based on the information specified in section 2.2.3 of this document:

```
\\MyMERXServer\MERXFileShare\status\generic\TestProductSetup\0\1.0.0.0\sample\Status.Txt
```

10. The MERX client attempts to read the Status.Txt file, and finds that no Status.Txt file exists.

11. Because no Policy.Txt and no Status.Txt file exist, this error is subject to the default "Crashes per bucket" value of 5. The MERX client checks to make sure that 5 error reporting files have not already been collected for this problem. It does this by looking at the Count.Txt file for the error:

```
\\MyMERXServer\MERFileShare\counts\generic\TestProductSetup\0\1.0.0.0\sample\Count.Txt
```

    The Count.Txt file has the following contents:

```
Cabs Gathered=3
Total Hits=17
```

    Because 3 is fewer than 5, the MERX client continues the data collection process.

1. Because there is no Status.Txt file, the MERX client does not add any additional data to the error report. The MERX client compresses the original report files into a single CAB-format file with the randomly generated name of "3tu58e7c.Cab".

2. The MERX client copies the error report file to the MERX file share:

```
\\MyMERXServer\MERXFileShare\cabs\generic\TestProductSetup\0\1.0.0.0\sample\3tu58e7c.Cab
```

12. The MERX client updates the following file on the MERX file share to increment the number of hits and the number of copied error reporting files.

```
\\MyMERXServer\MERXFileShare\counts\generic\TestProductSetup\0\1.0.0.0\sample\Count.Txt
```

    The Count.Txt file now has the following contents:

```
Cabs Gathered=4
Total Hits=18
```

# 5 Security

## 5.1 Security Considerations for Implementers

This section is as specified in [MS-CER] section 5.1.

## 5.2 Index of Security Parameters

This section is as specified in [MS-CER] section 5.2

# 6 Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs.

- The 2007 Microsoft Office system
- Microsoft Office 2010 suites

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that the product does not follow the prescription.

<1> Section 2.2.3: This list of blocked characters is because of the fact that the Param values can be used in SMB filepaths (as specified in [MS-SMB]) and in other scenarios, which each contribute restrictions on which characters may be used. The MERX client used by Office performs simple character substitutions (for example, the underscore "_" character in place of prohibited punctuation characters, or the letter "X" replacing the first letter of prohibited words) to bring the parameters into conformance.

<2> Section 2.2.3.2.5: Many products, including Office, now use Generic Error Reporting (as specified in section 2.2.3.4 of this document) to report installation failures since it allows each product to define an appropriate parameter set for its error reports.

# 7 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

# 8 Index

## S



## T



## V



## W