[MS-THCH]:
Tracing HTTP Correlation Header Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **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 may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft Open Specification Promise or the Community Promise. If you would prefer a written license, or if the technologies described in the Open Specifications 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 may 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 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 specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications do 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 are intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

Revision Summary

Date	Revision History	Revision Class	Comments
12/16/2011	1.0	New	Released new document.
03/30/2012	1.0	No change	No changes to the meaning, language, or formatting of the technical content.
07/12/2012	1.0	No change	No changes to the meaning, language, or formatting of the technical content.
10/25/2012	1.0	No change	No changes to the meaning, language, or formatting of the technical content.
01/31/2013	1.0	No change	No changes to the meaning, language, or formatting of the technical content.
08/08/2013	1.0	No change	No changes to the meaning, language, or formatting of the technical content.
11/14/2013	1.0	No change	No changes to the meaning, language, or formatting of the technical content.
02/13/2014	1.0	No change	No changes to the meaning, language, or formatting of the technical content.
05/15/2014	1.0	No change	No changes to the meaning, language, or formatting of the technical content.

Contents

1 Introduction

The Tracing HTTP Correlation Header Protocol specifies the E2EActivity HTTP header which can be used by an HTTP/1.1 client to communicate a unique identifier for an HTTP message to an HTTP server. The identifier is used in turn by the server to correlate traces generated by the server to messages received from the client.

Sections 1.8, 2, and 3 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in RFC 2119. Sections 1.5 and 1.9 are also normative but cannot contain those terms. All other sections and examples in this specification are informative.

1.1 Glossary

The following terms are defined in [MS-GLOS]:

base64
client
encoding
GUID
HTTP client
HTTP server
Hypertext Transfer Protocol (HTTP)

The following terms are specific to this document:

ETW: Event Tracing for Windows. For more information, see [MSDN-ETW].

Representational State Transfer (REST): A software architecture implementation for distributed hypermedia systems, such as the World Wide Web.

tracing: A mechanism used to write out diagnostic information.

WCF service: Windows Communication Foundation (WCF) service. A program that exposes a collection of endpoints for communicating with client applications or other service applications.

Windows Communication Foundation (WCF): A unified programming model for building service-oriented applications that enable development of secure, reliable, transacted solutions that integrate across platforms.

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

References to Microsoft Open Specifications documentation do not include a publishing year because links are to the latest version of the documents, which are updated frequently. References to other documents include a publishing year when one is available.

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.

[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

[RFC2616] Fielding, R., Gettys, J., Mogul, J., et al., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999, http://www.ietf.org/rfc/rfc2616.txt

1.2.2 Informative References

[MSDN-ETW] Microsoft Corporation, "Improving Debugging and Performance Tuning with ETW", http://msdn.microsoft.com/en-us/magazine/cc163437.aspx

[MSDN-WCF] Microsoft Corporation, "Windows Communication Foundation", http://msdn.microsoft.com/en-us/library/ms735119.aspx

[MSDN-WCFREST] Microsoft Corporation, "REST in Windows Communication Foundation (WCF)", http://msdn.microsoft.com/en-us/netframework/cc950529.aspx

[MS-GLOS] Microsoft Corporation, "Windows Protocols Master Glossary".

[SOAP1.1] Box, D., Ehnebuske, D., Kakivaya, G., et al., "Simple Object Access Protocol (SOAP) 1.1", May 2000, http://www.w3.org/TR/2000/NOTE-SOAP-20000508/

1.3 Overview

The Tracing HTTP Correlation Header Protocol specifies the E2EActivity HTTP header.

In **HTTP**/1.1, an **HTTP client** can specify a unique identifier for an HTTP message by including the E2EActivity HTTP header in the HTTP message. When the message is received by the **HTTP server**, the identifier can be used when emitting **traces** to provide a correlation between generated traces and incoming messages from the client.<1>

There are no changes to the HTTP messages sent from the server to the client based on receipt of the E2EActivity HTTP header.

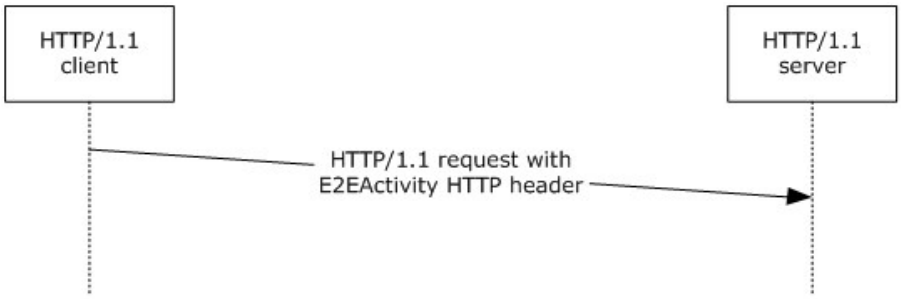

Figure 1: Sequence diagram showing communication of the E2EActivity HTTP header between the HTTP client and HTTP server

1.4 Relationship to Other Protocols

None.

1.5 Prerequisites/Preconditions

None.

1.6 Applicability Statement

When no other mechanism exists for an HTTP server to uniquely identify an HTTP message received from an HTTP client, the client can use the E2EActivity HTTP header to correlate the traces generated by the server in response to messages received from the client.

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

HTTP/1.1 is the only transport supported by this protocol for use of the E2EActivity HTTP header.

2.2 Message Syntax

The E2EActivity HTTP header defined by this protocol can be used by HTTP clients when sending HTTP/1.1 messages. The syntax for HTTP/1.1 messages is defined in [RFC2616].

To provide the unique identifier, the HTTP client SHOULD **base64**-encode the identifier as a **GUID** and include it as the value for the E2EActivity HTTP header in the HTTP header collection in the HTTP message. The client SHOULD specify a unique identifier value for each HTTP message it sends. The following example shows a typical E2EActivity header with a base64-encoded value:

```
E2EActivity: GWABtfYCDEu4hxOZR7sWGQ==
```

Upon receipt of the HTTP message from the client, the HTTP server SHOULD base64-decode the GUID value of the E2EActivity HTTP header in the HTTP message. The server MUST then include this identifier value when emitting traces for the corresponding HTTP message. By doing so, the server traces can be correlated to the received HTTP message which caused the trace to be generated.

3 Protocol Details

3.1 HTTP/1.1 Client Details

3.1.1 Abstract Data Model

None.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Higher-Layer Triggered Events

An HTTP/1.1 client can include the E2EActivity HTTP header (section 2.2) in the HTTP messages it sends to the HTTP server.

3.1.5 Message Processing Events and Sequencing Rules

When an HTTP/1.1 **client** includes the E2EActivity HTTP header in the HTTP messages it sends to the HTTP server, the response message from the server is not affected. Therefore, the client processing rules for response messages received from the server MUST NOT change.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

4 Protocol Examples

The following example shows how an HTTP/1.1 client specifies a base64-encoded unique identifier as the value for the E2EActivity HTTP header in the HTTP message. In this example, the GUID value "100f44d4-c7ac-45dc-98f7-974c064d61dd" is base64-encoded as "1EQPEKzH3EWY95dMBk1h3Q==" in the E2EActivity HTTP header in the HTTP message. When a value is specified for the E2EActivity HTTP header, the HTTP server includes the value when generating tracing data related to the received message.

```
POST http://server/Service/Service1.svc HTTP/1.1
Content-Type: text/xml; charset=utf-8
E2EActivity: 1EQPEKzH3EWY95dMBk1h3Q==
Content-Length: 157
```

5 Security

5.1 Security Considerations for Implementers

None.

5.2 Index of Security Parameters

None.

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:

- Microsoft .NET Framework 4.5

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 1.3: The Windows implementation of this protocol is exercised in **Windows Communication Foundation** [MSDN-WCF] when **ETW** tracing [MSDN-ETW] is enabled on the client and the client is communicating with a **WCF service** over the HTTP transport. In this scenario, common message exchange patterns can include **REST** [MSDN-WCFREST] and SOAP [SOAP1.1].

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