[MS-HTTPE]:
Hypertext Transfer Protocol (HTTP) Extensions

Revision Summary

Date	Revision History	Revision Class	Comments
11/14/2013	1.0	New	Released new document.
02/13/2014	2.0	Major	Significantly changed the technical content.

Contents

1 Introduction

This document specifies a set of extensions to the HyperText Transfer Protocol (HTTP), dealing with the internationalization of host names and query strings.

Sections 1.8, 2, and 3 of this specification are normative and contain RFC 2119 language. Section 1.5 and 1.9 are also normative but do not contain RFC 2119 language. All other sections and examples in this specification are informative.

1.1 Glossary

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

ASCII
code page
.NET Framework
Unicode
Uniform Resource Locator (URL)

The following terms are specific to this document:

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. We will assist you in finding the relevant information.

[MS-UCODEREF] Microsoft Corporation, "Windows Protocols Unicode Reference".

[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

[RFC3629] Yergeau, F., "UTF-8, A Transformation Format of ISO 10646", STD 63, RFC 3629, November 2003, http://www.ietf.org/rfc/rfc3629.txt

[RFC3986] Berners-Lee, T., Fielding, R., and Masinter, L., "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005, http://www.ietf.org/rfc/rfc3986.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

[RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, August 2010, http://rfc-editor.org/rfc/rfc5890.txt

[TR46] Davis, M., and Suignard, M., "Unicode IDNA Compatibility Processing", Unicode Technical Standard #46, September 2012, http://www.unicode.org/reports/tr46/

1.2.2 Informative References

[ISO/IEC-8859-1] International Organization for Standardization, "Information Technology -- 8-Bit Single-Byte Coded Graphic Character Sets -- Part 1: Latin Alphabet No. 1", ISO/IEC 8859-1, 1998, http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=28245

Note There is a charge to download the specification.

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

[MSDN-WinINetRef] Microsoft Corporation, "WinINet Reference", http://msdn.microsoft.com/en-us/library/windows/desktop/aa385483(v=vs.85).aspx

[RFC2396] Berners-Lee, T., Fielding, R., and Masinter, L., "Uniform Resource Identifiers (URI): Generic Syntax", RFC 2396, August 1998, http://www.ietf.org/rfc/rfc2396.txt

[RFC6943] Thaler, D., Ed., "Issues in Identifier Comparison for Security Purposes", RFC 6943, May 2013, http://www.rfc-editor.org/rfc/rfc6943.txt

1.3 Overview

This document specifies a set of extensions to the HyperText Transfer Protocol (HTTP) [RFC2616], dealing with internationalization of host names and query strings, and also with the path syntax.

Originally the HTTP protocol was defined only in terms of the **ASCII** character set. However, there quickly became a demand to support languages other than English. Among other things, this notably affected two types of data. First, it affected strings passed in the "query" component of a **Uniform Resource Locator (URL)** (later called a Uniform Resource Identifier (URI) in [RFC3986]) for use in fields in forms, doing lookups in search engines, and so on. Second, a demand arose to give servers names in the native language, thus resulting in an internationalized host name.

A mechanism known as Internationalizing Domain Names in Applications (IDNA), for supporting internationalized host names in protocols defined for ASCII, was later standardized and is now specified in [RFC5890] and [TR46]. In the meantime, the extensions in this document were already in use, which include:

- The transport of query string parameters in URIs without being percent-encoded.
- The use of characters in the HTTP Host header without being limited to the ASCII subset of characters, as opposed to requiring IDNA encoding to get an ASCII string to include.

A second extension is that the syntax of the path component of a URI is extended to allow "[" and "]" without being percent encoded.

1.4 Relationship to Other Protocols

This document specifies extensions to **HTTP**, and retains the same relationships to other protocols as the base HTTP protocol does. For encoding formats, the query extension in this document is an alternative to the encoding format specified in [RFC2616] and the Host header extension in this document is an alternative to the IDNA encoding format.

1.5 Prerequisites/Preconditions

These extensions assume that the client and the server have both been configured to use the same **code page**.

1.6 Applicability Statement

The extensions in this document are applicable only to environments where clients and servers all use the same code page. Furthermore, they are also applicable only to environments where either no HTTP proxy is present between the client and the server, or any HTTP proxies support the more liberal URI syntax defined in this document.

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

Messages are transported as specified in [RFC2616].

2.2 Message Syntax

The syntax is as specified in [RFC2616], except as follows.

2.2.1 Request-URI

The URI requested appears in the **Request-URI** field as specified in [RFC2616] section 5.1.2. It used the syntax restrictions in [RFC2396], which specifies that the query component of a URI can use only the ASCII subset of characters and requires other characters to be percent-escaped as specified in [RFC2396] section 2.4.1.

Note Although [RFC2396] was later obsoleted by [RFC3986], that restriction is unchanged. Specifically, [RFC3986] section 3.4 states:

```
query       = *( pchar / "/" / "?" )
```

This specification extends the Augmented Backus-Naur Form (ABNF) [RFC5234] for the query portion of the Request-URI field as follows:

```
query       = *(<any CHAR except CTLs or "#">)
```

Furthermore, [RFC3986] section 3.3 specifies the syntax of the path component and states:

```
pchar       = unreserved / pct-encoded / sub-delims / ":" / "@"
```

This specification extends this syntax to allow the "[" and "]" characters as follows:

```
pchar       = unreserved / pct-encoded / sub-delims / ":" / "@" / "[" / "]"
```

2.2.2 Host Header

HTTP is defined in [RFC2616] as using text encoded in ISO-8859-1 [ISO/IEC-8859-1]. The Host header is specified in [RFC2616] section 14.23 with a more restricted syntax, however. It uses the syntax restrictions specified in [RFC2396], which specifies that the Host header value can use only a limited set of characters, all within the ASCII character set.

This specification extends the Host header syntax to permit the value to be encoded in UTF-8 [RFC3629] or in the client's code page rather than requiring the use of IDNA to generate an ASCII string. This means that characters might be encoded using octets that are not allowed in ISO-8859-1.

3 Protocol Details

3.1 Client Details

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.

CodePage: The ANSI codepage that the client is configured to use. See [MS-UCODEREF] section 2.2.1 for more details.

3.1.2 Timers

None beyond what is specified in [RFC2616].

3.1.3 Initialization

None beyond what is specified in [RFC2616].

3.1.4 Higher-Layer Triggered Events

3.1.4.1 Sending an HTTP Request

When a higher-layer protocol or application requests the content for a given URI, the HTTP implementation MUST construct the HTTP request as specified in [RFC2616], except as follows.

Characters not legal in the standard query syntax SHOULD<1> be escaped as specified in [RFC2396] section 2.4.1 but MAY instead be encoded (unescaped) in the configured **CodePage** as specified in [MS-UCODEREF] section 3.1.5.1.1.2.

Characters that are not legal in the standard Host header syntax SHOULD be encoded by using the IDNA algorithm as specified in [RFC5890] and [TR46], but MAY<2> instead be encoded in UTF-8 [RFC3629] or encoded in the configured **CodePage** as specified in [MS-UCODEREF] section 3.1.5.1.1.2.

3.1.5 Message Processing Events and Sequencing Rules

None beyond what is specified in [RFC2616].

3.1.6 Timer Events

None beyond what is specified in [RFC2616].

3.1.7 Other Local Events

None beyond what is specified in [RFC2616].

3.2 Server Details

3.2.1 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.

CodePage: The ANSI codepage that the server is configured to use. See [MS-UCODEREF] section 2.2.1 for more details.

3.2.2 Timers

None beyond what is specified in [RFC2616].

3.2.3 Initialization

None beyond what is specified in [RFC2616].

3.2.4 Higher-Layer Triggered Events

None beyond what is specified in [RFC2616].

3.2.5 Message Processing Events and Sequencing Rules

3.2.5.1 Receiving an HTTP Request

When an HTTP Request message is received, the HTTP server MUST validate it and process it as specified in [RFC2616], except as follows.

If the Request-URI contains a query component that does not conform to the standard syntax, but does conform to the extended syntax specified in section 2.2.1, the server MUST interpret it as being encoded in the server's **CodePage** (see [MS-UCODEREF] section 3.1.5.1.1.3 for details). If the Request-URI contains a query component containing a percent (%) character, the server SHOULD interpret it as being escaped as specified in [RFC2396] section 2.4.1. The server MAY<3> instead interpret it literally; that is, where the percent character represents itself rather than indicating the beginning of an escape sequence.

The Host header MUST be validated using the extended syntax specified in section 2.2.2. If the value contains characters that would not be valid in the standard syntax, the server SHOULD interpret it as follows: attempt to interpret it as **UTF-8** and if it is not a valid UTF-8 sequence, then interpret it in as being encoded in the server's **CodePage** (see [MS-UCODEREF] section 3.1.5.1.1.3 for details). The server MAY<4> instead reverse the order of checks; that is, first attempt to interpret it as being encoded in the server's **CodePage** and if it is not a valid string in that **CodePage**, then interpret it as being encoded in UTF-8.

3.2.6 Timer Events

None beyond what is specified in [RFC2616].

3.2.7 Other Local Events

None beyond what is specified in [RFC2616].

4 Protocol Examples

In the following examples, the client and server are both configured to use the ANSI Baltic code page (code page 1257), and an application requests that http://bønne.contoso.com/path?søster be retrieved. Both the host component and the query component of this URI contain a LATIN SMALL LETTER O WITH STROKE ("ø" which is **Unicode** U+00F8). Note that in neither example does it appear encoded in ISO-8859-1 (that is, with octet value 248 = 0xF8) as HTTP would normally require for all headers.

4.1 Request Sent Through a Proxy

In this example, the request is being sent through a proxy, so the Request-URI includes a hostname. Since the host and query portions of the URI both contain a non-ASCII character, the client chooses to use the extended syntax and the HTTP request appears as follows (possibly along with other HTTP headers).

```
GET http://xn--bnne-gra.contoso.com/?s%C3%B8ster HTTP/1.1
Host: bønne.contoso.com
```

In this request, the LATIN SMALL LETTER O WITH STROKE is encoded as follows.

- In the host component of the URI in the request line, it appears in the host name's IDNA form (xn--bnne-gra) as in normal HTTP without any extensions.
- In the query component of the URI in the request line, it appears in the escaped form of the UTF-8 encoding (U+00F8 encoded in UTF-8 is 0xC3 0xB8) as in normal HTTP without any extensions.
- In the Host header, the client chooses to use the ANSI Baltic code page (octet value 184 = 0xB8) instead of the IDNA form. As such, other HTTP utilities might misinterpret the "ø" as being (in ISO-8859-1) a CEDILLA ("¸" which is Unicode U+00B8) and display it as "b¸nne.contoso.com".

4.2 Request Not Sent Through a Proxy

In this example, the request is not sent through a proxy, so the Request-URI does not contain a hostname. Since the host and query portions of the URI both contain a non-ASCII character, the client chooses to use the extended syntax and the HTTP request appears as follows (possibly along with other HTTP headers).

```
GET /?søster HTTP/1.1
Host: bønne.contoso.com
```

In this request, the LATIN SMALL LETTER O WITH STROKE is encoded as follows.

- In the query component of the URI in the request line, it appears encoded in the ANSI Baltic code page (octet value 184 = 0xB8). As such, other HTTP utilities might misinterpret the "ø" as being (in ISO-8859-1) a CEDILLA ("¸" which is Unicode U+00B8) and display it as "s¸ster".
- In the Host header, the client chooses to use UTF-8 and encodes the ø as 0xC3 0xB8. As such, other HTTP utilities might misinterpret the "ø" as being (in ISO-8859-1) a LATIN CAPITAL LETTER A WITH TILDE ("Ã" which is Unicode U+00C3) followed by CEDILLA ("¸" which is Unicode U+00B8) and display it as "bÃ¸nne.contoso.com".

5 Security

5.1 Security Considerations for Implementers

Security considerations are discussed in [RFC2616] section 15. Since the query component and Host header are often used for comparison against expected strings, and since the extensions in this document allow additional ways to encode the same strings, care should be taken to ensure that matching algorithms operate correctly, typically by normalizing a string to some common form before comparison. For further discussion of security considerations for comparison algorithms, see [RFC6943].

5.2 Index of Security Parameters

None.

6 Appendix A: Product Behavior

This document specifies version-specific details in the Microsoft .NET Framework. For information about which versions of .NET Framework are available in each released Windows product or as supplemental software, see **.NET Framework**.

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 2.0
- Microsoft .NET Framework 3.5
- Microsoft .NET Framework 4.0
- Microsoft .NET Framework 4.5
- Windows 2000 operating system
- Windows XP operating system
- Windows Server 2003 operating system
- Windows Server 2003 R2 operating system
- Windows Vista operating system
- Windows Server 2008 operating system
- Windows Server 2008 R2 operating system
- Windows 7 operating system
- Windows 8 operating system
- Windows Server 2012 operating system
- Windows 8.1 operating system
- Windows Server 2012 R2 operating system

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 3.1.4.1: Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows Server 2008 R2, Windows 7, Windows 8, and Windows Server 2012 use the ANSI code page with the extended query syntax.

<2> Section 3.1.4.1: Windows uses UTF-8 for the Host header by default when sending to destinations in the Intranet zone. .NET Framework 2.0, .NET Framework 3.5, .NET Framework 4.0,

and .NET Framework 4.5 use the ANSI code page for the Host header value by default. When the "idn" configuration flag is enabled, the .NET Framework uses the IDNA form.

<3> Section 3.2.5.1: Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows Server 2008 R2, Windows 7, Windows 8, and Windows Server 2012 do not decode the sequence but return the string directly to the higher-layer protocol or application. It is thus the responsibility of the higher-layer protocol or application to determine how to interpret the string.

<4> Section 3.2.5.1: Windows allows the order to be configured.

7 Change Tracking

This section identifies changes that were made to the [MS-HTTPE] protocol document between the November 2013 and February 2014 releases. Changes are classified as New, Major, Minor, Editorial, or No change.

The revision class **New** means that a new document is being released.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements or functionality.
- The removal of a document from the documentation set.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **Editorial** means that the formatting in the technical content was changed. Editorial changes apply to grammatical, formatting, and style issues.

The revision class **No change** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the technical content of the document is identical to the last released version.

Major and minor changes can be described further using the following change types:

- New content added.
- Content updated.
- Content removed.
- New product behavior note added.
- Product behavior note updated.
- Product behavior note removed.
- New protocol syntax added.
- Protocol syntax updated.
- Protocol syntax removed.
- New content added due to protocol revision.
- Content updated due to protocol revision.
- Content removed due to protocol revision.
- New protocol syntax added due to protocol revision.
- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.

- Obsolete document removed.

Editorial changes are always classified with the change type **Editorially updated.**

Some important terms used in the change type descriptions are defined as follows:

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.
- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

The changes made to this document are listed in the following table. For more information, please contact dochelp@microsoft.com.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
1.3 Overview	70774 Added that the specification includes extending the syntax of the path component.	Y	Content updated.
1.6 Applicability Statement	70774 Specified the applicability of the extension when an HTTP proxy is present.	Y	Content updated.
2.2.1 Request-URI	70774 Described the extension to the syntax of the path component.	Y	Content updated.

8 Index