[MS-PCCRTP]:
Peer Content Caching and Retrieval:
Hypertext Transfer Protocol (HTTP) Extensions

Revision Summary

Date	Revision History	Revision Class	Comments
12/05/2008	0.1	Major	Initial Availability
01/16/2009	0.1.1	Editorial	Revised and edited the technical content.
02/27/2009	0.1.2	Editorial	Revised and edited the technical content.
04/10/2009	0.1.3	Editorial	Revised and edited the technical content.
05/22/2009	0.2	Minor	Updated the technical content.
07/02/2009	1.0	Major	Updated and revised the technical content.
08/14/2009	2.0	Major	Updated and revised the technical content.
09/25/2009	3.0	Major	Updated and revised the technical content.
11/06/2009	4.0	Major	Updated and revised the technical content.
12/18/2009	4.1	Minor	Updated the technical content.
01/29/2010	4.1.1	Editorial	Revised and edited the technical content.
03/12/2010	4.1.2	Editorial	Revised and edited the technical content.
04/23/2010	4.1.3	Editorial	Revised and edited the technical content.
06/04/2010	4.1.4	Editorial	Revised and edited the technical content.
07/16/2010	4.1.4	No change	No changes to the meaning, language, or formatting of the technical content.
08/27/2010	4.1.4	No change	No changes to the meaning, language, or formatting of the technical content.
10/08/2010	4.1.4	No change	No changes to the meaning, language, or formatting of the technical content.
11/19/2010	4.1.4	No change	No changes to the meaning, language, or formatting of the technical content.
01/07/2011	4.1.4	No change	No changes to the meaning, language, or formatting of the technical content.
02/11/2011	4.1.4	No change	No changes to the meaning, language, or formatting of the technical content.
03/25/2011	4.1.4	No change	No changes to the meaning, language, or formatting of the technical content.
05/06/2011	4.1.4	No change	No changes to the meaning, language, or formatting of the technical content.

Date	Revision History	Revision Class	Comments
06/17/2011	4.2	Minor	Clarified the meaning of the technical content.
09/23/2011	4.3	Minor	Clarified the meaning of the technical content.
12/16/2011	5.0	Major	Significantly changed the technical content.
03/30/2012	5.0	No change	No changes to the meaning, language, or formatting of the technical content.
07/12/2012	5.0	No change	No changes to the meaning, language, or formatting of the technical content.
10/25/2012	6.0	Major	Significantly changed the technical content.
01/31/2013	6.0	No change	No changes to the meaning, language, or formatting of the technical content.
08/08/2013	7.0	Major	Significantly changed the technical content.

Contents

1 Introduction

The Peer Content Caching and Retrieval: HTTP Extensions specify a new content encoding, **PeerDist**, that can be used in HTTP/1.1. In particular, this protocol specifies the mechanism used by an HTTP/1.1 client and an HTTP/1.1 server to communicate to each other using the PeerDist content encoding.

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]:

HTTP client
Hypertext Transfer Protocol (HTTP)
Transmission Control Protocol (TCP)

The following terms are specific to this document:

Accept-Encoding: The **HTTP** header that defines the type of **content coding** that the client will accept from the server as part of the **HTTP** response. See [RFC2616] section 14.3 for details.

content coding: The type of encoding transformation that has been applied or can be applied to an **entity-body**. See [RFC2616] section 3.5 for details.

Content-Encoding: The **HTTP** header that defines the types of **content coding** that have been applied to the **HTTP entity-body**. See [RFC2616] section 14.11 for details.

entity-body: The name given to the payload of an **HTTP** request or response. See [RFC2616] section 1.3 for details.

FIN: The **TCP** control bit that signals no more data from sender. See [RFC793] section 3.2 for details.

PeerDist: The name of the new content encoding specified in this document.

PeerDist Content Encoding: A way of presenting an **HTTP** entity-body (defined in [RFC2616]) through its metadata, in the form of a Content Information Data Structure, as defined in [MS-PCCRC] section 2.3, which is derived from the content using algorithms described in [MS-PCCRC] sections 2.1 and 2.2.

PeerDist Content Information: A Message-body (defined in [RFC2616]) obtained for the requested content using **PeerDist Content Encoding**. Identifies the data produced by running one of the algorithms specified in [MS-PCCRC] section 2.3 on a given input.

PeerDist-encoded response entity body: See **PeerDist Content Information**.

PeerDist-Supporting Request: A request sent by a client that is able to participate in peer content caching and retrieval. A PeerDist-Supporting Request will have its Accept-Encoding header value set to PeerDist.

RST: The **TCP** control bit that signals a connection reset. See [RFC793] section 3.2 for details.

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.

A reference marked "(Archived)" means that the reference document was either retired and is no longer being maintained or was replaced with a new document that provides current implementation details. We archive our documents online [Windows Protocol].

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. Please check the archive site, http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624, as an additional source.

[MS-PCCRC] Microsoft Corporation, "Peer Content Caching and Retrieval: Content Identification".

[RFC793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981, http://www.ietf.org/rfc/rfc0793.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

[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

1.2.2 Informative References

[MC-BUP] Microsoft Corporation, "Background Intelligent Transfer Service (BITS) Upload Protocol".

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

1.3 Overview

HTTP/1.1 is a client/server-based protocol. The purpose of PeerDist content encoding is to enable peer content caching and retrieval in HTTP/1.1.

Using PeerDist content encoding, allows an **HTTP/1.1 client** to participate in the peer content caching and retrieval process. Upon detecting PeerDist encoding support from a client, an **HTTP/1.1 server** that supports peer content caching may choose to send a PeerDist-encoded response. The message body (that is, an encoded entity body) of such a response takes the form of the Content Information Data Structure as specified in [MS-PCCRC] section 2.3, constructed for the requested content using the algorithms described in [MS-PCCRC] sections 2.1 and 2.2. Receiving a PeerDist-encoded response allows an HTTP/1.1 client to use the information present in the response to discover and download actual content from peers.

A typical PeerDist-encoded response is orders of magnitude smaller than a response that is not PeerDist encoded; the actual content transfer occurs between peers. Thus, PeerDist content encoding can reduce the burden of distributing the content from the HTTP/1.1 server.

A sequence diagram describing the communication between an HTTP/1.1 client and the HTTP/1.1 server is shown here.

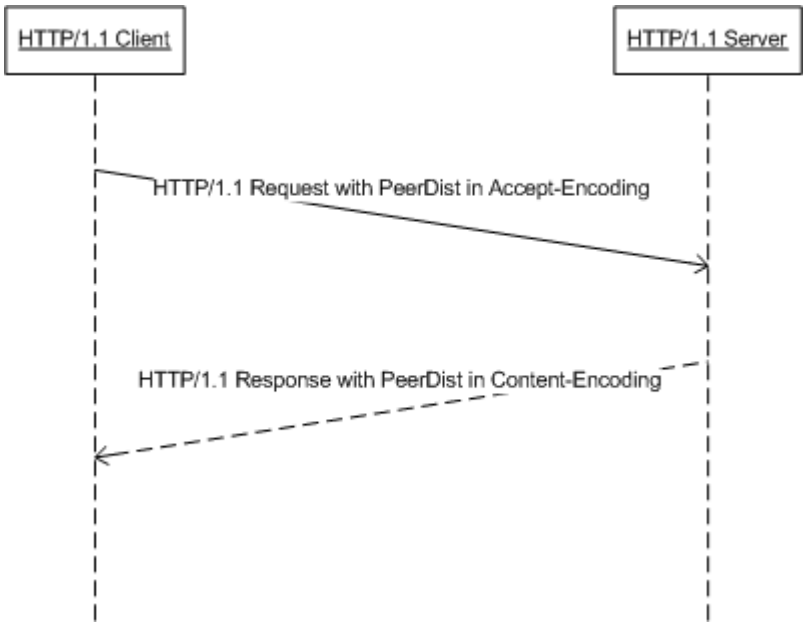

Figure 1: Sequence diagram describing the communication between the HTTP/1.1 client and the HTTP/1.1 server

1.4 Relationship to Other Protocols

The PeerDist content encoding defined in this document is intended to be used for HTTP/1.1.

The PeerDist content encoding is used by clients and servers that are capable of participating in peer content caching and retrieval.

The PeerDist content encoding uses the Content Information data structure defined in [MS-PCCRC] section 2.3.

1.5 Prerequisites/Preconditions

None.

1.6 Applicability Statement

Advertising PeerDist content encoding capability is applicable for an HTTP/1.0 client or HTTP/1.1 client (only) if it is able to participate in peer content caching and retrieval.<1>

Using PeerDist content encoding is applicable for an HTTP/1.1 server (only) when communicating to an HTTP/1.1 client that has advertised its capability to participate in peer content caching and retrieval.

1.7 Versioning and Capability Negotiation

The PeerDist content encoding defined in this document uses a version parameter that the HTTP/1.1 client sets to specify the maximum version of PeerDist content encoding that the client supports.<2>

The PeerDist content encoding defined in this document uses a version parameter that the HTTP/1.1 server sets to specify the version of PeerDist content encoding that is used for the HTTP response.<3>

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

This document defines PeerDist, a new content encoding that can be used in HTTP/1.1. HTTP/1.1 is the transport for all messages used by the PeerDist content encoding.

2.2 Message Syntax

HTTP/1.1 [RFC2616] defines the syntax of HTTP/1.1 messages.

This document defines a new content encoding value, namely PeerDist. The PeerDist content-encoding value can be specified in the **Accept-Encoding** and **Content-Encoding** header fields, as shown in the following examples.

```
Accept-Encoding: gzip, deflate, peerdist
Content-Encoding: peerdist
```

In addition, this document also defines two new extension-header field values. The syntax of these header field values is described as follows.

```
extension-header = X-P2P-PeerDist
X-P2P-PeerDist = "X-P2P-PeerDist" ":" peerdist-params
X-P2P-PeerDistEx = "X-P2P-PeerDistEx" ":" peerdistex-params
```

The X-P2P-PeerDist and X-P2P-PeerDistEx extension-header fields can appear in both requests and responses. The purpose of these header fields is to carry additional parameters when the PeerDist content encoding is used.

```
peerdist-params = 1#( version | [content-len] | [missing-data-request] )
version = "Version" "=" major-version "." minor-version
major-version = 1*DIGIT
minor-version = 1*DIGIT
```

Note that there can be no spaces between major-version and "." as well as "." and minor-version. The major and minor versions MUST be considered as separate multidigit numbers. Thus, version 1.23 is higher than version 1.3.

The *Version* parameter is used by the HTTP/1.1 client to specify the maximum version of PeerDist content encoding that the client supports. The *Version* parameter is used by the HTTP/1.1 server to specify the version of PeerDist content encoding that was used for the response.

```
content-len = "ContentLength" "=" 1*DIGIT
```

The *content-len* parameter contains the length of the **entity-body**, defined in [RFC2616] section 1.3, in octets, before the PeerDist content encoding is applied to it.

The *missing-data-request* parameter is used by the HTTP/1.1 client and is set to true to indicate to the server that the client is sending the request because it was unable to retrieve data from its peers. This parameter MUST NOT be specified when the PeerDist content encoding is specified in the Accept-Encoding header field value.

```
missing-data-request = "MissingDataRequest" "=" ( "true" )
```

The *peerdistex-params* parameter is used by the HTTP/1.1 client to indicate to the server which versions of the PeerDist Content Information Data Structure, as specified in [MS-PCCRC] section 2.3, the client supports. *MinContentInformation* is always equal to 1.0 and indicates support for version 1.0 of the PeerDist Content Information Data Structure. If *MaxContentInformation* is set to 1.0, then the client only supports version 1.0 of the PeerDist Content Information structure, but if *MaxContentInformation* is set to 2.0, then the client also supports version 2.0 of the PeerDist Content Information Data Structure.

```
peerdistex-params = 1#( "MinContentInformation=1.0, MaxContentInformation=" ( "1.0" | "2.0" )
| [make-hash-request] | [hash-request] )
```

The *make-hash-request* parameter is used by the HTTP/1.1 server to indicate to the client to make a hash request for the content that the client requested because the hashes were not available with the server at the time of the request.

```
make-hash-request = ", MakeHashRequest" "=" ( "true" )
```

The *hash-request* parameter is used by the HTTP/1.1 client to indicate to the server that this is a hash request for the content which the client previously requested. This parameter is used in a hash request to the server when the server sends a data response with the **MakeHashRequest** field set to true.

```
hash-request = ", HashRequest" "=" ( "true" )
```

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 MAY<4> include the PeerDist content encoding in its Accept-Encoding header field value for every HTTP request it sends, as shown in the following example.

```
Accept-Encoding: gzip, deflate, peerdist
```

If the client chooses to use the PeerDist content encoding for an HTTP request, the client MUST also include the PeerDist parameters header field in the same HTTP request. As shown in the following example, the PeerDist parameters header field MUST contain the *Version* parameter containing the highest version of the PeerDist content encoding that the client supports.

```
X-P2P-PeerDist: Version=1.0
```

If the PeerDist parameters header field contains a *Version* parameter equal to 1.1, then the client MUST also include a PeerDistEx parameters header field which MUST include *MinContentInformation* and *MaxContentInformation* parameters indicating the minimum and maximum version of the PeerDist Content Information structure that the client supports.

```
X-P2P-PeerDistEx: MinContentInformation=1.0, MaxContentInformation=2.0
```

An HTTP/1.0 client MAY<5> include the PeerDist content encoding in the Accept-Encoding header field value of its HTTP requests.

3.1.5 Message Processing Events and Sequencing Rules

3.1.5.1 Receiving a Response of a PeerDist-Supporting Request

When an HTTP/1.1 client sends an HTTP request with the PeerDist content encoding listed in its Accept-Encoding header, the HTTP/1.1 server MAY send an HTTP response with a "Connection" header field with a value of "close". When an HTTP/1.1 client receives such a response, it SHOULD close the underlying **TCP** connection gracefully by sending an **FIN** instead of an **RST**.

If the response from the server contains a **PeerDistEx** parameters header field with *MakeHashRequest* set to true, then the client SHOULD make a hash request to the server and include the **PeerDistEx** parameters header field with *HashRequest* set to true.

```
X-P2P-PeerDistEx: MinContentInformation=1.0, MaxContentInformation=2.0, HashRequest=true
```

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

3.2 HTTP/1.1 Server Details

When the HTTP/1.1 request indicates that the client supports the PeerDist content encoding, then if the response contains an **ETag** header field, a **Last-Modified** header field, or both header fields, the HTTP/1.1 server MAY<6> use the PeerDist content encoding. [RFC2616] section 14.11 defines content encoding usage.

The HTTP/1.1 server MAY use the PeerDist content encoding in its response to an HTTP/1.0 request if the HTTP/1.0 request includes an Accept-Encoding header field containing PeerDist.

3.2.1 Abstract Data Model

None.

3.2.2 Timers

None.

3.2.3 Initialization

None.

3.2.4 Higher-Layer Triggered Events

None.

3.2.5 Message Processing Events and Sequencing Rules

The server constructs, for the requested content, a Content Information Data Structure defined in [MS-PCCRC] section 2.3 using the algorithms described in [MS-PCCRC] sections 2.1 and 2.2 and places such a structure in the response message as an encoded entity body.

3.2.5.1 Receiving a PeerDist-Supporting Request

If the HTTP/1.1 server uses the PeerDist content encoding for its response, then the server MUST construct for the requested content, a Content Information Data Structure as specified in [MS-PCCRC] section 2.3, using the algorithms described in [MS-PCCRC] sections 2.1 and 2.2, and place such a structure in the response message as an encoded entity-body.

If the X-P2P-PeerDistEx header is present, the server MUST generate and respond with a Content Information Data Structure whose version falls within the range specified by the *MinContentInformation* and *MaxContentInformation* parameters. If the values of *MinContentInformation* and *MaxContentInformation* do not fall within the range specified in section 2.2, the server MUST not generate and respond with a Content Information Data Structure, and MUST respond with another client-supported encoding as defined in [RFC2616]. If no X-P2P-PeerDistEx extension header was present, then the server MUST respond with a version 1.0 Content Information Data Structure.

It MUST also include the PeerDist parameters header field in the response. The PeerDist parameters header field MUST contain the *Version* parameter containing the version of the PeerDist content encoding used in the response. As shown in the following example, the PeerDist parameters header field MUST also contain the *ContentLength* parameter specifying the content length of the response entity-body before the PeerDist content encoding has been applied to it.

```
Content-Encoding: PeerDist
X-P2P-PeerDist: Version=1.0, ContentLength=102400
```

If the HTTP/1.1 server sends a PeerDist-encoded response entity-body, it MUST encode the entity-body into segments and blocks as specified in [MS-PCCRC] section 2, and then use that encoding to construct a Content Information Data Structure, as specified in [MS-PCCRC] section 2.3. It MUST then use this latter data structure as the PeerDist-encoded response entity-body.

If the HTTP/1.1 server does not have the Content Information Data Structure available for the content requested by the client, for such reasons as this is the first request for the content, then the server SHOULD send a response containing the original content and add the X-P2P-PeerDistEx header with *MakeHashRequest* set to true. This indicates to the client to make an additional request for the content hashes.

The HTTP/1.1 server MAY<7> choose to use the algorithms and data structures defined in [MS-PCCRC] on the response entity-body before sending it to the HTTP/1.1 client. Furthermore, it MAY<8> send the "Connection" header field with a value of "close" to require the HTTP/1.1 client not to use the same connection for future HTTP requests. The HTTP/1.1 server SHOULD NOT<9> send the "Connection" header field in its response if the HTTP/1.1 client is known to be unable to handle the "Connection" header field gracefully, as specified in section 3.1.5.1.

3.2.6 Timer Events

None.

3.2.7 Other Local Events

None.

4 Protocol Examples

When the HTTP client uses the PeerDist content encoding, it specifies PeerDist in the Accept-Encoding header field, as shown in the following example.

```
GET /index.html HTTP/1.1
Host: www.hostname.com
Accept: */*
Accept-Language: en-US
Accept-Encoding: gzip, deflate, peerdist
X-P2P-PeerDist: Version=1.1
X-P2P-PeerDistEx: MinContentInformation=1.0, MaxContentInformation=1.0
User-Agent: Mozilla/4.0
```

In this example, the HTTP client announces that it is ready to accept the response entity-body that is encoded using the PeerDist content encoding. It also declares the version of the PeerDist content encoding for which it is configured, as well as the minimum and maximum Content Information Data Structure versions it supports.

If the server sends the HTTP response entity-body encoded with PeerDist **content coding**, then it will set the Content-Encoding header field value to peerdist as shown in the following example.

```
HTTP/1.1 200 OK
Content-Type: text/html
Content-Encoding: peerdist
Content-Length: 198
Last-Modified: Fri, 01 Aug 2008 01:02:03 GMT
Accept-Ranges: bytes
ETag: "8d2babfc81f3c81"
Server: Microsoft-IIS/7.0
X-P2P-PeerDist: Version=1.1, ContentLength=184946
Date: Fri, 01 Aug 2008 10:20:30 GMT
...198 bytes of PeerDist Content Information...
```

In this response, the server indicates that the content is encoded using the PeerDist content encoding. The server used version 1.0 of the PeerDist content encoding. The server could not generate version 2.0 of the PeerDist content encoding because the client specified a *MaxContentInformation* parameter equal to 1.0. Had the client specified a *MaxContentInformation* parameter equal to 2.0, then the server could have chosen to respond with version 2.0 of the PeerDist content encoding. The server also includes the content length of the entity-body when it is encoded using the identity content coding. In other words, the **Content-Length** header field would have had the value 184946 if the Content-Encoding header was either missing or specified "identity" as defined in [RFC2616].

If the server does not have the Content Information Data Structure at the time of the request, the server responds with the original content and includes the X-P2P-PeerDistEx header with *MakeHashRequest* set to true as shown in the following example.

```
HTTP/1.1 200 OK
Content-Length: 184946
Content-Type: image/png
Last-Modified: Thu, 31 Mar 2011 20:17:35 GMT
Accept-Ranges: bytes
ETag: "c184b9ace0efcb1:0"
```

```
Server: Microsoft-IIS/8.0
X-P2P-PeerDist: Version=1.1
X-P2P-PeerDistEx: MakeHashRequest=true
```

In response to the above message, the client sends a hash request with the X-P2P-PeerDistEx header and *HashRequest* set to true as shown in the following example.

```
GET /welcome.png HTTP/1.1
Host: www.example.com
X-P2P-PeerDist: Version=1.1
X-P2P-PeerDistEx: MinContentInformation=1.0, MaxContentInformation=2.0, HashRequest=true
```

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:

- Windows Vista operating system
- Windows Server 2008 operating system
- Windows 7 operating system
- Windows Server 2008 R2 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 1.6: For Windows Vista and Windows Server 2008, support for the client-side elements of this protocol is available only via the optional installation of the Background Intelligent Transfer Service (see [MC-BUP]) via Windows Management Framework. Support for the server-side elements of this protocol is not available for Windows Vista or Windows Server 2008.

<2> Section 1.7: HTTP/1.1 clients in Windows Vista, Windows Server 2008, Windows 7, and Windows Server 2008 R2 set the PeerDist version parameter to 1.0. HTTP/1.1 clients in Windows 8, Windows Server 2012, Windows 8.1, and Windows Server 2012 R2 set the PeerDist version parameter to 1.1.

<3> Section 1.7: HTTP/1.1 servers in Windows Server 2008 R2 set the PeerDist version parameter to 1.0. HTTP/1.1 servers in Windows Server 2012 and Windows Server 2012 R2 set the PeerDist version parameter to 1.1 when responding to a client that specified a PeerDist version parameter equal to 1.1 and set the PeerDist version parameter to 1.0 when replying to a client that specified a PeerDist version parameter equal to 1.0.

<4> Section 3.1.4: HTTP/1.1 clients in Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, and Windows Server 2012 R2 use the PeerDist content encoding for GET requests only.

<5> Section 3.1.4: HTTP/1.0 clients in Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, and Windows Server 2012 R2 use the PeerDist content encoding for GET requests only.

<6> Section 3.2: In Windows Server 2008 R2, Windows Server 2012, and Windows Server 2012 R2, the HTTP/1.1 server sends a PeerDist-encoded response.

<7> Section 3.2.5.1: The HTTP/1.1 server in Windows Server 2008 R2 uses the algorithms and data structures defined in [MS-PCCRC] to generate the **PeerDist Content Information** only when it receives an HTTP/1.1 request. The server runs the algorithms asynchronously, and therefore it does not use the PeerDist encoding for the response to the request that triggered the execution of the algorithms. Similarly, the server does not use the PeerDist encoding for any HTTP/1.1 requests for the same content that are received during the execution of the algorithms on that content. However, after the algorithms have completed and the PeerDist Content Information has been generated for that content, the server will respond to requests for the same content with the PeerDist Content Information for that content.

<8> Section 3.2.5.1: The HTTP/1.1 server in Windows Server 2008 R2 sends the "Connection" header field with a value of "close" if the HTTP request is a range retrieval request, and the total length of the full entity-body is greater than 1 megabyte.

<9> Section 3.2.5.1: The HTTP/1.1 server in Windows Server 2008 R2 does not send the "Connection" header field with a value of "close" if the HTTP/1.1 client is "Microsoft BITS".

7 Change Tracking

This section identifies changes that were made to the [MS-PCCRTP] protocol document between the January 2013 and August 2013 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.
- An extensive rewrite, addition, or deletion of major portions of content.
- The removal of a document from the documentation set.
- Changes made for template compliance.

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 language and 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 or language changes were introduced. The technical content of the document is identical to the last released version, but minor editorial and formatting changes, as well as updates to the header and footer information, and to the revision summary, may have been made.

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.
- New content added for template compliance.
- Content updated for template compliance.
- Content removed for template compliance.
- 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 protocol@microsoft.com.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
2.2 Message Syntax	67973 Updated the syntax for the MakeHashRequest and HashRequest parameters.	Y	Content updated.
4 Protocol Examples	67973 Updated the example syntax regarding when the server does not have the Content Information Data Structure at the time of the request.	Y	Content updated.
6 Appendix A: Product Behavior	Modified this section to include references to Windows 8.1 operating system and Windows Server 2012 R2 operating system.	Y	Content updated.

8 Index

T

V