

[MS-BPCR]: Background Intelligent Transfer Service (BITS) Peer- Caching: Content Retrieval Protocol Specification

Intellectual Property Rights Notice for Protocol Documentation

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

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

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

Revision Summary

Date	Revision History	Revision Class	Comments
04/10/2007	1.0		Version 1.0 release
05/18/2007	1.2		Version 1.2 release
06/08/2007	1.2.1	Editorial	Revised and edited the technical content.

Date	Revision History	Revision Class	Comments
07/10/2007	1.3	Minor	Updated the technical content.
08/17/2007	1.3.1	Editorial	Revised and edited the technical content.
09/21/2007	1.4	Minor	Updated the technical content.
10/26/2007	1.4.1	Editorial	Revised and edited the technical content.
01/25/2008	1.4.2	Editorial	Revised and edited the technical content.

Table of Contents

1	Introduction	6
1.1	Glossary	6
1.2	References	6
1.2.1	Normative References	6
1.2.2	Informative References	7
1.3	Protocol Overview (Synopsis)	7
1.4	Relationship to Other Protocols	8
1.5	Prerequisites/Preconditions	8
1.6	Applicability Statement	8
1.7	Versioning and Capability Negotiation	9
1.8	Vendor-Extensible Fields	9
1.9	Standards Assignments.....	9
2	Messages	10
2.1	Transport	10
2.2	Message Syntax	10
2.2.1	Common Data Types.....	10
2.2.1.1	guid	10
2.2.1.2	url	11
2.2.1.3	searchStatus	11
2.2.1.4	fileRange	12
2.2.1.5	cacheRecord.....	12
2.2.1.6	searchRequest.....	13
2.2.1.7	searchResponse.....	14
2.2.2	DISCOVERY-REQUEST.....	14
2.2.2.1	Standard HTTP Header Fields.....	14
2.2.2.2	HTTP Header Fields	14
2.2.2.3	Message Body	14
2.2.3	DISCOVERY-RESPONSE.....	14
2.2.3.1	HTTP Headers.....	15
2.2.3.2	Body Data.....	15
2.2.4	DOWNLOAD-REQUEST	15
2.2.5	DOWNLOAD-RESPONSE	15
2.2.6	HEAD-REQUEST	16
2.2.7	HEAD-RESPONSE.....	16
3	Protocol Details	17
3.1	Client Details	17
3.1.1	Abstract Data Model.....	17
3.1.1.1	Table of Servers	17
3.1.1.2	Call	17
3.1.1.3	Search.....	18
3.1.1.3.1	STATE_INIT	19
3.1.1.3.2	STATE_CHOOSE_SERVER	19
3.1.1.3.3	STATE_SEND_REQUEST	20
3.1.1.3.4	STATE_WAIT	20
3.1.1.3.5	STATE_DISCOVER_SERVERS	20
3.1.1.3.6	STATE_COMPLETE	20
3.1.1.4	Download Request	20
3.1.2	Timers	21
3.1.2.1	Search Timeout	21
3.1.2.2	Call Request Timeout.....	21

3.1.2.3	Download Request Timeout	21
3.1.3	Initialization	21
3.1.4	Higher-Layer Triggered Events	21
3.1.4.1	New Search Request	21
3.1.4.2	Cancel a Search in Progress	21
3.1.4.3	New Download Request	21
3.1.5	Message Processing Events and Sequencing Rules	21
3.1.5.1	Call Response	21
3.1.5.2	Download Response	22
3.1.6	Timer Events.....	22
3.1.6.1	Call Response Timeout.....	22
3.1.6.2	Download Response Timeout	22
3.1.6.3	Search Timeout	22
3.1.7	Other Local Events.....	22
3.1.7.1	Call Events.....	22
3.1.7.1.1	Problem with Server Certificate During a Call	22
3.1.7.1.2	Connection Failure During a Call	23
3.1.7.2	Download Events	23
3.1.7.2.1	Problem with Server Certificate During a Download	23
3.1.7.2.2	Connection Failure During Download	23
3.1.7.3	Search Events	23
3.1.7.3.1	A Pending Call Completes	23
3.1.7.3.2	RESULT_FOUND	23
3.1.7.3.3	RESULT_NOT_FOUND	23
3.1.7.3.4	RESULT_CLIENT_CERT_UNKNOWN	23
3.1.7.3.5	RESULT_ACCESS_DENIED or RESULT_INVALID_SEARCH or RESULT_UNKNOWN	23
3.1.7.3.6	RESULT_SERVER_CERT_UNKNOWN	24
3.1.7.3.7	RESULT_TRANSPORT_ERROR or RESULT_OUT_OF_RESOURCES.....	24
3.1.7.3.8	Notification of New Server or Address	24
3.1.7.3.9	Protocol Shutdown.....	24
3.2	Server Details.....	24
3.2.1	Abstract Data Model.....	24
3.2.1.1	Table of Cache Records.....	24
3.2.2	Timers	25
3.2.3	Initialization.....	25
3.2.4	Higher-Layer Triggered Events	25
3.2.4.1	Protocol Shutdown	25
3.2.5	Message Processing Events and Sequencing Rules	25
3.2.5.1	General Rules for HTTP-Level Error Responses.....	25
3.2.5.2	Message Validation.....	25
3.2.5.3	DISCOVERY-REQUEST	26
3.2.5.4	DOWNLOAD-REQUEST	26
3.2.5.5	HEAD-REQUEST	27
3.2.6	Timer Events.....	27
3.2.7	Other Local Events.....	27
4	Protocol Example.....	28
4.1	Successful Search with Two Servers	28
5	Security	34
5.1	Security Considerations for Implementers	34
5.2	Index of Security Parameters	34
6	Appendix A: Windows Behavior	35

7	Appendix B: XML Schema.....	38
8	Index.....	40

1 Introduction

This document is a specification for the Background Intelligent Transfer Service (BITS) Peer-Caching: Content Retrieval Protocol. This Microsoft-proprietary protocol is one of the family of protocols that implements a distributed URL cache known as "BITS peer-caching". Other protocols in the family are used to discover potential **peers** and to authenticate them. A client uses the BITS Peer-Caching: Content Retrieval Protocol to search an existing set of peers for content and to download from those peers.

1.1 Glossary

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

Domain
Extended Key Usage (EKU)
Globally Unique Identifier (GUID)
Security Identifier (SID)
Unicode

The following terms are specific to this document:

Header Field: As specified in [\[RFC2616\]](#) section 4.2.

Message Body: As specified in [\[RFC2616\]](#) section 4.3.

Peer: A single device or node in a **peer-to-peer** networking system.

Peer-to-Peer: A serverless networking technology that allows several participating network devices to share resources and communicate directly with each other.

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as specified 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. Please check the archive site, <http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624>, as an additional source.

[IANAPORT] Internet Assigned Numbers Authority, "Port Numbers", November 2006, <http://www.iana.org/assignments/port-numbers>

[MS-GLOS] Microsoft Corporation, "[Windows Protocols Master Glossary](#)", March 2007.

[MS-ERREF] Microsoft Corporation, "[Windows Error Codes](#)", January 2007.

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

[RFC2246] Dierks, T. and Allen, C., "The TLS Protocol Version 1.0", RFC 2246, January 1999, <http://www.ietf.org/rfc/rfc2246.txt>

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

[RFC2781] Hoffman, P. and Yergeau, F., "UTF-16, an encoding of ISO 10646", RFC 2781, February 2000, <http://www.ietf.org/rfc/rfc2781.txt>

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000, <http://www.ietf.org/rfc/rfc2818.txt>

[RFC3280] Housley, R., Polk, W., Ford, W., and Solo, D., "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3280, April 2002, <http://www.ietf.org/rfc/rfc3280.txt>

[RFC3339] Klyne, G. and Newman, C., "Date and Time on the Internet: Timestamps", RFC 3339, July 2002, <http://www.ietf.org/rfc/rfc3339.txt>

[XML] World Wide Web Consortium, "Extensible Markup Language (XML) 1.0 (Fourth Edition)", W3C Recommendation, September 2006, <http://www.w3.org/TR/REC-xml>

1.2.2 Informative References

[MS-BPAU] Microsoft Corporation, "[Background Intelligent Transfer Service \(BITS\) Peer-Caching: Peer Authentication Protocol Specification](#)", April 2007.

[MS-BPDP] Microsoft Corporation, "[Background Intelligent Transfer Service \(BITS\) Peer-Caching: Peer Discovery Protocol Specification](#)", April 2007.

[MSDN-BITS] Microsoft Corporation, "Background Intelligent Transfer Service", <http://msdn2.microsoft.com/en-us/library/aa362827.aspx>

1.3 Protocol Overview (Synopsis)

The Background Intelligent Transfer Service (BITS) Peer-Caching: Content Retrieval Protocol defines methods for a network client both to query multiple servers for data associated with a given URL and to download that data.

In Windows, the BITS component uses the BITS Peer-Caching: Content Retrieval Protocol to implement a distributed **peer-to-peer** cache of data items based on associated HTTP and HTTPS URLs as well as UNC paths. BITS discovers peer servers by using the [Background Intelligent Transfer Service \(BITS\) Peer-Caching: Peer Discovery Protocol](#) (for more information, see [MS-BPDP]) and authenticates them by using the [Background Intelligent Transfer Service \(BITS\) Peer-Caching: Peer Authentication Protocol](#) (for more information, see [MS-BPAU]). For more information on BITS, see [\[MSDN-BITS\]](#).

The BITS Peer-Caching: Content Retrieval Protocol does not address issues of cache management such as policies for adding and removing content or the method of storing and indexing the content. These issues are internal to the server implementation.

To start, the client uses some other protocol to identify a URL containing a data image to be downloaded. The client uses another protocol, such as the Background Intelligent Transfer Service (BITS) Peer-Caching: Peer Discovery Protocol (for more information, see [MS-BPDP]), to identify hosts that may be serving some or all of the image data via the BITS Peer-Caching: Content Retrieval Protocol. The client chooses several such hosts and queries them concurrently by using the BITS Peer-Caching: Content Retrieval Protocol to see if one or more can serve the image data. Based on the responses, the client chooses whether to download the image data from the hosts and/or the original URL. For this algorithm, see section [3.1.1.3](#).

1.4 Relationship to Other Protocols

The Background Intelligent Transfer Service (BITS) Peer-Caching: Content Retrieval Protocol is a client/server protocol that uses HTTPS over TLS 1.0 as its transport. A host that implements the client side or server side of this protocol typically also implements the [Background Intelligent Transfer Service \(BITS\) Peer-Caching: Peer Discovery Protocol](#) (for more information, see [MS-BPDP]) and the [Background Intelligent Transfer Service \(BITS\) Peer-Caching: Peer Authentication Protocol](#) (for more information, see [MS-BPAU]) to automate the location and authentication of servers.

The consumer of this protocol may be either a top-level application or another client/server protocol.

The following diagram shows the hierarchy of protocols on which content retrieval depends.

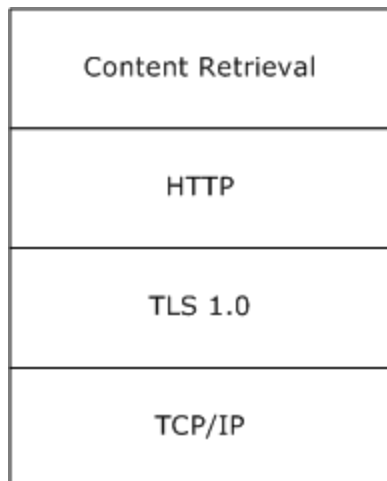


Figure 1: Protocol hierarchy for BITS

1.5 Prerequisites/Preconditions

A client of the Background Intelligent Transfer Service (BITS) Peer-Caching: Content Retrieval Protocol **MUST** have identified a set of potential servers by an implementation-dependent method. [<1>](#)

Computers in both the client and server roles **MUST** be provisioned with certificates accessible to the HTTPS protocol, as specified in [\[RFC2246\]](#) sections 7.4.2 and 7.4.3. The client and server **MUST** agree on the requirements for certificates. [<2>](#)

For the protocol to be effective, servers **MUST** be provisioned with a cache of URL content. Rules for population, expiration, and so forth are not addressed in this protocol.

1.6 Applicability Statement

Because the BITS Peer-Caching: Content Retrieval Protocol uses unicast communication to poll multiple servers for content, it is best suited for situations in which the client is connected to the servers by a high-speed network.

The BITS Peer-Caching: Content Retrieval Protocol is more complex than the standard HTTP proxy behavior specified in [\[RFC2616\]](#) section 8.1.3, but it does not require the server to download data on the client's behalf. The BITS Peer-Caching: Content Retrieval Protocol is best suited to a peer-to-

peer environment in which the client may choose among several servers based on connection speed, authentication decisions, and other factors.

1.7 Versioning and Capability Negotiation

The BITS Peer-Caching: Content Retrieval Protocol does not define an explicit system for version negotiation. The presence of individual capabilities is implicitly signaled in each message by the presence or absence of optional fields. For details of each message, see section [2.2](#).

1.8 Vendor-Extensible Fields

The BITS Peer-Caching: Content Retrieval Protocol uses HRESULTS, as specified in [\[MS-ERREF\]](#), primarily in [DISCOVERY-REQUEST \(section 3.2.5.3\)](#). Vendors are free to choose their own values as long as the C bit (0x20000000) is set, indicating it is a customer code.

1.9 Standards Assignments

Parameter	Value	Reference
TCP port for HTTPS listener	2178	As specified in [IANAPORT] .

2 Messages

The following sections specify how BITS Peer-Caching: Content Retrieval Protocol messages are transported and BITS Peer-Caching: Content Retrieval Protocol common data types.

2.1 Transport

Messages MUST be transported over HTTPS by using port 2178.

The client and server MUST supply a certificate to the HTTPS protocol (as specified in [RFC2818](#) section 2.2.1) for transmission to their interlocutors. The certificates used MUST be within their validity interval when the connection is initiated.

A client or server MAY impose additional requirements on the certificate for authentication purposes.[<3>](#)

2.2 Message Syntax

Messages follow HTTP/1.1 syntax. The required HTTP headers and the format of the HTTP **message body** for each message are specified in the following sections. An implementation MAY include additional HTTP headers in each message, following the rules specified in [RFC2616](#) section 2.2, and MUST treat recognized headers according to their standard meaning specified in [RFC2616](#) section 4.2.[<4>](#)

A future version of the BITS Peer-Caching: Content Retrieval Protocol may define new HTTP **header fields** and XML elements. The recipient of a message MUST ignore header fields and XML elements it does not understand.

2.2.1 Common Data Types

The [DISCOVERY-REQUEST](#) message and the [DISCOVERY-RESPONSE](#) message rely on XML (as specified in [XML](#)). The following table shows the standard XML namespaces used within the BITS Peer-Caching: Content Retrieval Protocol and the alias (prefix) used in the remaining sections of this protocol specification.

Alias (prefix)	XML namespace
s	http://www.w3.org/2001/XMLSchema

The following table shows the Microsoft-defined XML namespace used within the BITS Peer-Caching: Content Retrieval Protocol and the alias (prefix) used in the remaining sections of this protocol specification.

Alias (prefix)	XML namespace
cd	http://schemas.microsoft.com/windows/2007/01/BITS/ContentDiscovery

The following sections list the elements defined in this namespace.

2.2.1.1 guid

A **globally unique identifier (GUID)** of an object or entity within the protocol.

```

<s:simpleType name="guid">
  <s:restriction base="s:string">
    <s:pattern value="[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-
      [0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}" />
  </s:restriction>
</s:simpleType>

```

2.2.1.2 url

A URL string. URLs within the BIT) Peer-Caching: Content Retrieval Protocol are limited to a maximum of 2,200 characters.

```

<simpleType name="url">
  <restriction base="string">
    <maxLength value="2200" />
  </restriction>
</simpleType>

```

2.2.1.3 searchStatus

The status code for a search request:

```

<simpleType name="searchStatus">
  <restriction base="string">
    <enumeration value="Success"/>
    <enumeration value="CertificateNotFound"/>
    <enumeration value="ContentNotFound"/>
    <enumeration value="AccessDenied"/>
    <enumeration value="OutOfResources"/>
    <enumeration value="InvalidSearch"/>
    <enumeration value="Unknown"/>
  </restriction>
</simpleType>

```

Value	Meaning
Success	The server found one or more cache records matching the search criteria.
ContentNotFound	The server holds no cache records matching the search criteria.
OutOfResources	The search request could not be processed due to a transient error.
InvalidSearch	The server did not understand the given search criteria, or the given criteria are not allowed.
CertificateNotFound	The client's certificate is syntactically correct but not known to the server. If an out-of-band authentication protocol such as the BITS Peer-Caching: Peer Authentication Protocol (for more information, see [MS-BPAU]) is used, the client should retry the request after authenticating. Otherwise, it is equivalent to "AccessDenied".
AccessDenied	The client is forbidden from downloading from this server.
Unknown	An uncategorized error occurred.

2.2.1.4 fileRange

The description of a contiguous range of data within a URL:

```
<complexType name="fileRange">
  <sequence>
    <element name="Offset" type="unsignedLong"/>
    <element name="Length" type="unsignedLong"/>
  </sequence>
</complexType>
```

Offset: Location of the beginning of the range in bytes relative to the start of the URL data.

Length: Length of the range in bytes.

2.2.1.5 cacheRecord

The description of a record within the server's cache database:

```
<complexType name="cacheRecord">
  <sequence>
    <element name="Id" type="cd:guid"/>
    <element name="CreationTime" type="dateTime"/>
    <element name="ModificationTime" type="dateTime"/>
    <element name="LastAccessTime" type="dateTime"/>
    <element name="OriginUrl" type="cd:url"/>
    <element name="LocalUrl" type="cd:url"/>
    <element name="FileModificationTime" type="dateTime"/>
    <element name="FileSize" type="unsignedLong"/>
    <element name="FileEtag" type="string" minOccurs="0" />
    <element name="ContentRange" type="cd:fileRange"
      maxOccurs="unbounded"/>
    <any minOccurs="0" maxOccurs="unbounded"
      processContents="lax" namespace="##other"/>
  </sequence>
</complexType>
```

Id: The unique ID of the record.

OriginUrl: The URL being cached by the record.

CreationTime: The UTC time of creation of the record.

LastAccessTime: The UTC time of last access to the record.

ModificationTime: The UTC time of last modification to the record.

LocalUrl: The URI of the data in the record relative to the hostname and port of the server. The format (specified in [RFC2616](#) section 3.2.2) is:

```
Local_url = abs_path ["?" query]
```

FileModificationTime: The UTC modification time of the URL.

FileSize: The length in bytes of the URL content.

FileEtag: The HTTP entity tag of the URL, as specified in [\[RFC2616\]](#) section 3.11.[<5>](#)

ContentRange: One such element is returned, in order, for each range of bytes that is present in the cache record. For example, if the record contained 100 bytes at offset 2,000 followed by 100 bytes at offset 3,000, the returned XML would include the following <ContentRange> data:

```
<ContentRange>
  <Offset>2000</Offset>
  <Length>100</Length>
</ContentRange>
<ContentRange>
  <Offset>3000</Offset>
  <Length>100</Length>
</ContentRange>
```

A cache record encompassing the entire URL is represented as a single range with offset zero and length equal to the **FileSize** element.

2.2.1.6 searchRequest

A query to a single server for a single URL:

```
<complexType name="searchRequest">
  <sequence>
    <element name="OriginUrl" type="cd:url"/>

    <element name="FileModificationTime" type="dateTime" />
    <element name="FileSize" type="unsignedLong" minOccurs="0"/>
    <element name="FileEtag" type="string" minOccurs="0" />

    <element name="MaxRecords" type="positiveInteger"
      minOccurs="0" default="1" />

    <any minOccurs="0" maxOccurs="unbounded" processContents="lax"
      namespace="##other"/>
  </sequence>
</complexType>
```

OriginUrl: The URL for which the client is searching. The maximum length is 2,200 characters.

FileModificationTime: The UTC time stamp of the URL.

FileSize: The size in bytes of the URL.

FileEtag: The entity tag for the URL.

MaxRecords: The maximum number of records that may be included in the "searchResponse" element in the reply.[<6>](#) The server's response MUST abide by the limit, and the client SHOULD ignore response records beyond the limit.[<7>](#) If this element is omitted, there is no explicit limit on the number of records returned.

2.2.1.7 searchResponse

The result of a search request:

```
<complexType name="searchResponse">
  <sequence>
    <element name="Status" type="cd:searchStatus" />
    <element name="CacheRecord" type="cd:cacheRecord"
      minOccurs="0" maxOccurs="unbounded"/>
    <any minOccurs="0" maxOccurs="unbounded"
      processContents="lax" namespace="##other"/>
  </sequence>
</complexType>
```

2.2.2 DISCOVERY-REQUEST

The client sends a DISCOVERY-REQUEST to a server to inquire whether the server has cached a particular URL. The message is encoded as an HTTP POST request to the following URL:

/BITS-peer-caching

The request includes a number of fields in the HTTP message header. Some of them are standard fields (as specified in [\[RFC2616\]](#) section 4.5) that are required to take on specific values, while others are new fields defined by the BITS Peer-Caching: Content Retrieval Protocol. The fields MUST follow the rules defined in [\[RFC2616\]](#) section 4.2.

2.2.2.1 Standard HTTP Header Fields

Content-Length: The size in bytes of the HTTP message body. This field MUST be present.

2.2.2.2 HTTP Header Fields

X-ETW-ACTIVITY: A GUID-encoded activity correlation ID. An activity ID is a GUID that uniquely identifies the discovery request. The client MAY include this header as an aid to logging, enabling correlation between a client activity and the server activity. [<8>](#)

2.2.2.3 Message Body

The HTTP message body MUST be a **Unicode** XML 1.0 document that uses <http://schemas.microsoft.com/windows/2007/01/BITS/ContentDiscovery> as its default XML namespace. The document MUST use the UTF-8 or UTF-16 encoding; either byte ordering is allowed. The document MUST contain a "searchRequest" element. The recipient MAY choose to ignore element attributes. [<9>](#) The message body MUST NOT include additional data before or after the XML document. The XML document MAY contain trailing whitespace as part of the encoded content, as specified in [\[XML\]](#) section 2.1.

A server MUST support a maximum body size of at least 16 KB.

2.2.3 DISCOVERY-RESPONSE

The DISCOVERY-RESPONSE message is the response to a [DISCOVERY-REQUEST](#) message. It contains the results of the search—either an error or a set of matching cache records.

The HTTP status code MUST be 200. The following sections specify additional requirements.

2.2.3.1 HTTP Headers

Content-Length: MUST be the size in bytes of the HTTP message body. This field MUST be present.

2.2.3.2 Body Data

The HTTP message body MUST be a Unicode XML 1.0 document that uses <http://schemas.microsoft.com/windows/2007/01/BITS/ContentDiscovery> as its default XML namespace. The document MUST use the UTF-8 or UTF-16 encoding; either byte ordering is allowed. The document MUST contain a "searchResults" element. The recipient MAY choose to ignore element attributes. [<10>](#) The message body MUST NOT include additional data before or after the XML document. The XML document MAY contain trailing whitespace as part of the encoded content, as specified in [\[XML\]](#) section 2.1.

To allow for a large number of returned file ranges, a client SHOULD support a maximum response size of at least 1,024 KB. The XML MAY include comment tags to aid in readability. [<11>](#)

If the value of the "Status" child element is "Success", one or more CacheRecord elements MUST be present.

2.2.4 DOWNLOAD-REQUEST

To download data from a server, the client sends a DOWNLOAD-REQUEST, which is encoded as an HTTP GET request, as specified in [\[RFC2616\]](#) section 9.3. The request specifies the record ID and the requested range(s) within the record.

The URL MUST be specified as follows:

```
"/BITS-peercaching/{ " record-ID " }
```

where record-ID is the GUID ID of the record being requested, as returned in the **/SearchResults/CacheRecord/Id** element of a [DISCOVERY-RESPONSE](#).

A client MAY request a fraction of the record data by including a *Content-Range* header, as specified in [\[RFC2616\]](#) section 14.16. [<12>](#) If so, the requested ranges apply to the data in the record, not to the original URL data. For example, if a record contains bytes 100 to 199 of the URL, "Content-Range: 0-1 / 100" refers to bytes 100 and 101 of the original URL.

2.2.5 DOWNLOAD-RESPONSE

A DOWNLOAD-RESPONSE message is a standard HTTP/1.1 response packet. The HTTP status code MUST be either 200 or 206.

If the request contained a *Content-Range* header for a single range covering the entire record, the server MAY return either status 200 or 206. [<13>](#) Otherwise, the server MUST return status 206 when the request contains a *Content-Range* header and status 200 when the request does not contain a *Content-Range* header.

The response MUST include the *Content-Length* header. The format of the field is specified in [\[RFC2616\]](#) section 14.13.

The reply MAY include the *BITS_BASIC_INFO* header to provide finer-grained file time stamps.[<14>](#14) It provides the file's creation time, last access time, last modification time, and some file attributes in the following format:

```
BasicInfo = "0x" CreationTime ",0x" LastAccessTime ",0x"  
LastWriteTime ",0x" ChangeTime ",0xv" Attributes
```

All elements are required. Each of the following: *CreationTime*, *LastAccessTime*, *LastWriteTime*, and *ChangeTime* is a hexadecimal 64-bit integer representing a time in Windows FILETIME format. *Attributes* is a hexadecimal 32-bit integer representing a set of attribute flags supported by the FAT file system. Only FILE_ATTRIBUTE_ARCHIVE, FILE_ATTRIBUTE_HIDDEN, FILE_ATTRIBUTE_READONLY, and FILE_ATTRIBUTE_SYSTEM are allowed to be set; other flags MUST be set to zero, and MUST be ignored by the recipient.

2.2.6 HEAD-REQUEST

A client MAY request the attributes of a record without downloading data by sending a HEAD-REQUEST.[<15>](#15) It is encoded as a HEAD request; otherwise, the format is the same as a [DOWNLOAD-REQUEST \(section 2.2.4\)](#).

2.2.7 HEAD-RESPONSE

Following standard procedure for HEAD requests in HTTP, the reply to a [HEAD-REQUEST](#) is a reply that is identical to the reply for the equivalent [DOWNLOAD-REQUEST](#), except that a reply with status 200 or 206 excludes any body data.

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 what is described in this document.

3.1.1.1 Table of Servers

The client maintains a table of servers. Each row of the table contains the following data:

- Server DNS name.
- A Boolean variable indicating if the server has been authenticated by external means. This allows for "lazy" authentication if it is supported by the method used for peer discovery. If no external authentication is enforced, authenticated may be set to true.

3.1.1.2 Call

The Call holds the state associated with a network request sent to a particular server. The client call is specified by a state machine with the following states.

State	Description
STATE_INIT	The initial state.
STATE_RECEIVE	The request was sent to the server, and the call waits for a response.
STATE_PARSE_RESPONSE	The server's response was received and needs to be parsed.
STATE_COMPLETE	The terminal state.

A call contains several state elements:

- A pointer to a row in the table of potential servers.
- The XML request data sent to the server.
- An abstract completion result, which may be one of the following values.

Status	Description
RESULT_FOUND	The server found one or more cache records that match the search criteria.
RESULT_NOT_FOUND	The server found no cache records that match the search criteria.
RESULT_ACCESS_DENIED	The client is not authorized to access the server.
RESULT_CLIENT_CERT_UNKNOWN	The client needs to be authenticated to the server.

Status	Description
RESULT_SERVER_CERT_UNKNOWN	The server needs to be authenticated to the client.
RESULT_OUT_OF_RESOURCES	The server is too busy to process the request.
RESULT_TRANSPORT_ERROR	A lower-layer transport encountered an error.
RESULT_INVALID_SEARCH	The syntax of the request was not acceptable to the server.
RESULT_UNKNOWN	A protocol error occurred.

3.1.1.3 Search

Search is a data element that encapsulates the state associated with a particular search request from the higher-level protocol. A search can be represented by a state machine with the following states.

State	Description
STATE_INIT	The initial state for the machine.
STATE_CHOOSE_SERVER	The search needs to choose a server so that it can send a request.
STATE_SEND_REQUEST	The search needs to send a request to a server.
STATE_WAIT	The search is waiting for responses to its requests.
STATE_DISCOVER_SERVERS	The search needs to locate more servers, if possible.
STATE_ERROR	The call has completed unsuccessfully; STATUS_CODE contains the error code.
STATE_COMPLETE	The call has completed successfully.

The search contains several state elements.

- The URL search criteria passed by the higher-level protocol.
- **PENDING-CALL-TABLE:** A collection of pending calls.
- **SERVERS-NEEDED:** The number of servers remaining to be chosen.
- **F_DISCOVERED:** A flag that is true if the search previously searched for more servers.
- **F_WAITING_FOR_DISCOVERY:** A flag that is true if the search is waiting for more servers to be discovered.
- **NEW_SERVER:** A server from the table of servers.

The client defines a constant IDEAL-SERVER-COUNT as a target for the number of servers to contact. IDEAL-SERVER-COUNT SHOULD be set to 10.

The following figure illustrates the possible state transitions.

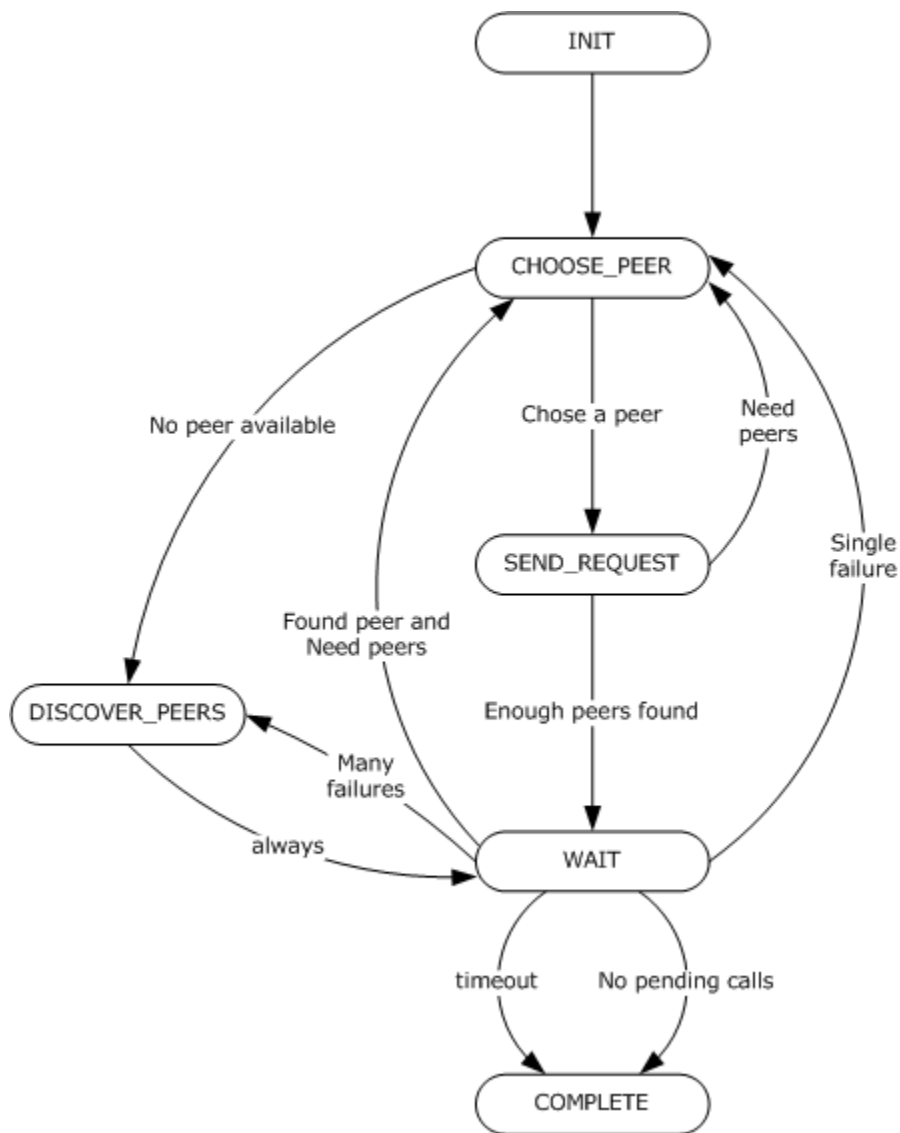


Figure 2: Possible state transitions

The actions taken at each state are described in the following sections.

3.1.1.3.1 STATE_INIT

Construct a [DISCOVERY-REQUEST](#) body from the supplied URL search criteria. Set F_DISCOVERED to false, and set F_WAITING_FOR_DISCOVERY to false. Set SERVERS-NEEDED to IDEAL-SERVER-COUNT. The search MAY initialize other states used to choose servers from the table. [<16>](#)

Set state to [STATE_CHOOSE_SERVER](#).

3.1.1.3.2 STATE_CHOOSE_SERVER

If F_DISCOVERED is false:

If (SERVERS-NEEDED > 0), choose another server (not chosen previously by this search) from the table of servers. A client MAY use external criteria when choosing a server, or may choose randomly from the table.<17>

If one was found, set NEW-SERVER to the server, decrement SERVERS-NEEDED, and set the state to [STATE_SEND_REQUEST](#); otherwise, set the state to [STATE_DISCOVER_SERVERS](#).

If F_DISCOVERED is true:

(SERVERS-NEEDED should be > 0.)

Set NEW_SERVER to a server (not chosen yet by this search) from the table. Decrement SERVERS-NEEDED, and set state to STATE_SEND_REQUEST.

3.1.1.3.3 STATE_SEND_REQUEST

Create a [Call](#) to NEW_SERVER, add it to the PENDING_CALL_TABLE, and send the request. If (SERVERS-NEEDED > 0), set state to [STATE_CHOOSE_SERVER](#); otherwise, set F_WAITING_FOR_DISCOVERY to false, and set state to [STATE_WAIT](#).

3.1.1.3.4 STATE_WAIT

Block, waiting for search timeout or a call to complete.

3.1.1.3.5 STATE_DISCOVER_SERVERS

If F_DISCOVERED is false, update the server list by implementation-dependent means.<18> Set F_DISCOVERED to true. If additional servers are discovered, notification will occur asynchronously, as specified in section [3.1.7.3.8](#).

Set state to [STATE_WAIT](#).

3.1.1.3.6 STATE_COMPLETE

This is the terminal state. Report the search status to the higher-layer protocol.

Note The conceptual data can be implemented by using a variety of techniques. Any data structure that stores the conceptual data may be used in the implementation.

3.1.1.4 Download Request

A download request represents a request from a higher-layer protocol for some or all of the data contained in a particular cache record. The request contains the following state elements:

HOST-ADDRESS: Host name or IP address of the server.

RECORD-ID: GUID of the cache record being downloaded.

RANGES: One or more byte ranges of the data in the cache record.

RESULT: An abstract completion result with the same range of values as the [Call](#) result.

3.1.2 Timers

3.1.2.1 Search Timeout

This timer limits the amount of time taken by any one search regardless of the state transitions involved. The default value is 60 seconds; the legal range is any positive value.

3.1.2.2 Call Request Timeout

This timer limits the amount of time that a [Call](#) waits for the response from the server. The default value is 15 seconds. An implementation MAY use a different value to accelerate detection of offline servers.

3.1.2.3 Download Request Timeout

This timer limits the amount of time that a Download request waits for the response from the server. The default SHOULD be at least 30 seconds. [<19>](#)

3.1.3 Initialization

When the client is initialized, it initializes the table of servers. It SHOULD begin listening for updates to the table of servers. [<20>](#)

3.1.4 Higher-Layer Triggered Events

3.1.4.1 New Search Request

The higher layer passes the URL search criteria.

The client instantiates a [Search](#) object with the associated URL data.

3.1.4.2 Cancel a Search in Progress

To cancel a search in progress, cancel each call in the PENDING-CALLS-TABLE.

3.1.4.3 New Download Request

To download cached data from a server, the higher-layer protocol passes the server, the cache record, and (optionally) one or more byte ranges to download. A new [Download](#) object is created.

3.1.5 Message Processing Events and Sequencing Rules

3.1.5.1 Call Response

As mentioned in [DISCOVERY-REQUEST \(section 2.2.2\)](#), a reply is considered a [DISCOVERY-RESPONSE](#) only if the HTTP status is set to 200. Any other HTTP status causes the call to be completed immediately. If the HTTP status is 503, the call result is [RESULT_OUT_OF_RESOURCES](#); otherwise, the call result is `RESULT_TRANSPORT_ERROR`.

When the HTTP status indicates success, the client parses the response's body. An error in body syntax sets the [Call](#) result to [RESULT_UNKNOWN](#).

The client then examines the `/SearchResults/Status` element of the body, and sets the call result according to the following table.

Element text	Result
"Success"	RESULT_FOUND
"ContentNotFound"	RESULT_NOT_FOUND
"AccessDenied"	RESULT_ACCESS_DENIED
"CertificateNotFound"	RESULT_CLIENT_CERT_UNKNOWN
"OutOfResources"	RESULT_OUT_OF_RESOURCES
"InvalidSearch"	RESULT_INVALID_SEARCH

3.1.5.2 Download Response

The response to a [DOWNLOAD-REQUEST](#) message is an HTTP reply. If the HTTP status is not 200, the download fails with [RESULT_TRANSPORT_ERROR](#), and the error is reported to the higher-layer protocol.

If the HTTP status is 200, the client validates the syntax of the response message. If it is invalid, the download fails with [RESULT_TRANSPORT_ERROR](#), and the error is reported to the higher-layer protocol.

If the message is valid, the download succeeds with [RESULT_FOUND](#). The result is reported to the higher-layer protocol along with the data from the cache record and the values from the [BITS_BASIC_INFO](#) header, if present.

3.1.6 Timer Events

3.1.6.1 Call Response Timeout

Complete the [Call](#) with [RESULT_TIMEOUT](#).

3.1.6.2 Download Response Timeout

Complete the download with [RESULT_TIMEOUT](#).

3.1.6.3 Search Timeout

Cancel each call in the [PENDING-CALLS-TABLE](#).

3.1.7 Other Local Events

3.1.7.1 Call Events

3.1.7.1.1 Problem with Server Certificate During a Call

During HTTPS connection setup, the [Call](#) judges the server's certificate by implementation-dependent criteria. [<21>](#) If the certificate fails validation, the client MAY attempt to authenticate the server by implementation-dependent means; if so, it completes the Call with the result [RESULT_SERVER_CERT_UNKNOWN](#). [<22>](#) Otherwise, the client completes the call with the result [RESULT_TRANSPORT_ERROR](#).

3.1.7.1.2 Connection Failure During a Call

The [Call](#) result is set to [RESULT_TRANSPORT_ERROR](#), and the Call is completed.

3.1.7.2 Download Events

3.1.7.2.1 Problem with Server Certificate During a Download

During HTTPS connection setup, the download judges the server's certificate by implementation-dependent criteria. [<23>](#) If the certificate fails validation, the client MAY attempt to authenticate the server by implementation-dependent means; if so, it completes the [Download](#) with the result [RESULT_SERVER_CERT_UNKNOWN](#). Otherwise, the client completes the download with the result [RESULT_TRANSPORT_ERROR](#).

3.1.7.2.2 Connection Failure During Download

The [Download](#) result is set to [RESULT_TRANSPORT_ERROR](#), and the Download is completed.

3.1.7.3 Search Events

3.1.7.3.1 A Pending Call Completes

Remove the call from the PENDING_CALLS_TABLE.

The client MAY change the status of the server in the server table based on the response. [<24>](#)

Take additional action based on the call's response status (see the following sections).

3.1.7.3.2 RESULT_FOUND

Report the received records to the higher-layer protocol. If the PENDING_CALLS_TABLE is empty, set state to [STATE_COMPLETE](#); otherwise, set state to [STATE_WAIT](#).

3.1.7.3.3 RESULT_NOT_FOUND

If the PENDING_CALLS_TABLE is empty, set state to [STATE_COMPLETE](#); otherwise, set state to [STATE_WAIT](#).

3.1.7.3.4 RESULT_CLIENT_CERT_UNKNOWN

Authenticate client certificate to server by implementation-dependent means. [<25>](#) If successful, create a new call, set NEW_SERVER to the call, and set state to [STATE_SEND_REQUEST](#).

Otherwise, remove the server from the server table, increment SERVERS_NEEDED, and set state to [STATE_CHOOSE_SERVER](#).

3.1.7.3.5 RESULT_ACCESS_DENIED or RESULT_INVALID_SEARCH or RESULT_UNKNOWN

Remove the server from the server table, increment SERVERS_NEEDED, and set state to [STATE_CHOOSE_SERVER](#).

3.1.7.3.6 RESULT_SERVER_CERT_UNKNOWN

Authenticate server certificate to client by implementation-dependent means. [<26>](#) If successful, create a new [Call](#), set NEW_SERVER to the server, and set state to [STATE_SEND_REQUEST](#).

Otherwise, remove the server from the server table, increment SERVERS_NEEDED, and set state to [STATE_CHOOSE_SERVER](#).

3.1.7.3.7 RESULT_TRANSPORT_ERROR or RESULT_OUT_OF_RESOURCES

Increment SERVERS_NEEDED, and set state to [STATE_CHOOSE_SERVER](#).

3.1.7.3.8 Notification of New Server or Address

Each search checks its F_WAITING_FOR_DISCOVERY flag and ignores the notification if the flag is false.

Otherwise, the search will initiate a new call to the server (using the newly discovered address, if applicable) and insert it into the search's table of pending calls. The search decrements SERVERS_NEEDED; if it is now zero, the F_WAITING_FOR_DISCOVERY flag is cleared.

3.1.7.3.9 Protocol Shutdown

Each pending [Search](#) object is canceled.

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 what is described in this document.

3.2.1.1 Table of Cache Records

The server maintains a table of the URL data in its cache. Each row of the table contains the following fields:

- GUID cache record ID.
- UTC time of creation of the cache record.
- UTC time of last modification to the cache record.
- UTC time of last access to the cache record.
- Origin URL.
- *Last-Modified* time stamp of the URL.
- Size in bytes of the URL.
- An ordered set of URL ranges cached in the record. If the record contains the entire URL content, a single range of byte 0 through (size in bytes) -1 is stored.

Note The preceding conceptual data can be implemented by using a variety of techniques. Any data structure that stores the preceding conceptual data may be used in this implementation.

3.2.2 Timers

No timers are required.

3.2.3 Initialization

The BITS Peer-Caching: Content Retrieval Protocol is initialized when a server wants to begin accepting client requests. The server begins listening for HTTPS connections on TCP port 2178 and instructs the HTTPS layer to require clients to provide a certificate.

3.2.4 Higher-Layer Triggered Events

3.2.4.1 Protocol Shutdown

When the BITS Peer-Caching: Content Retrieval Protocol is halted, the server stops processing new incoming messages and stops listening on the TCP port. The server MAY process existing messages or return a failure response. [.<27>](#)

3.2.5 Message Processing Events and Sequencing Rules

3.2.5.1 General Rules for HTTP-Level Error Responses

This section describes several circumstances in which the server's response to an incoming message is a response at the HTTP level rather than a message from section [2.2](#). In all such cases, the response MUST conform to the format specified in [\[RFC2616\]](#) section 6. The HTTP message body of these messages SHOULD be empty. [.<28>](#)

3.2.5.2 Message Validation

The server MUST validate the following aspects of a received message before determining the message type:

- HTTP version MUST be 1.1. [.<29>](#)
- HTTP verb must be one of the values in the first column of the table that follows.

If any of these checks fail, the server MUST reply with an HTTP error response, as specified in section [3.2.5.1](#), using an HTTP status code of 505.

HTTP verb	Message type
POST	DISCOVERY-REQUEST
GET	DOWNLOAD-REQUEST
HEAD	HEAD-REQUEST

Once the initial validation has succeeded, the server uses the HTTP verb to determine the message type, and processes the message as appropriate. For specific actions for each message type, see the following sections.

The server MAY impose limits on the number of messages processed simultaneously. [<30>](#) If an incoming message surpasses the server limits, the server SHOULD reply with an HTTP error response, as specified in section [2.2](#), using an HTTP status code of 503.

3.2.5.3 DISCOVERY-REQUEST

The server MAY use the client's certificate for authentication purposes. [<31>](#) If the server requires the client to authenticate itself by an out-of-band method before the certificate is accepted, the server MUST return a [DISCOVERY-RESPONSE](#) with the value of the SearchResults/Status element set to "CertificateNotFound".

If the URL of the message is not "/BITS-peer-caching", the server MUST reply with an HTTP error response, as specified in section [3.2.5.1](#), using an HTTP status code of 404.

The server validates the message in the following ways:

- The request MUST contain a Content-Length header.
- The value of the Content-Length header MUST be greater than zero and even.

If one or more of these checks fails, the server MUST reply with an HTTP error response, as specified in section [3.2.5.1](#), using an HTTP status code of 400.

The server MAY impose a limit on the size of the HTTP headers and the XML body. [<32>](#) The limit on the XML body MUST be at least 16 kilobytes.

The server then checks the XML body for syntactic correctness. A parsing error causes the server to return a DISCOVERY-RESPONSE containing only the <SearchResults> and <SearchResults>/<Status> elements, with the <Status> element describing the error category. For the allowable values of <Status>, see section [2.2.1.3](#).

After the message is validated, the server searches the table of cache records for records that match all of the provided criteria. Such records are placed in an unordered list; the list is truncated (if necessary) to conform to the provided <MaxRecords> value.

If the list is empty, the server sends a DISCOVERY-RESPONSE message with the <SearchResults>/<Status> element set to "ContentNotFound". Otherwise, the server sends a DISCOVERY-RESPONSE message with the <SearchResults>/<Status> element set to "Success"; the message contains one <SearchResults>/<CacheRecord> element for each record in the truncated list. [<33>](#)

3.2.5.4 DOWNLOAD-REQUEST

The server MAY use the client's certificate for authentication purposes. [<34>](#) If the server requires the client to authenticate itself by an out-of-band method before the certificate is accepted, the server MUST return an HTTP error response, as specified in section [3.2.5.1](#), using an HTTP status code of 400.

The URL of the message MUST be of the form:

```
"/BITS-peer-caching/{ " guid " }
```

If not, the server MUST reply with an HTTP error response, as specified in section [3.2.5.1](#), using an HTTP status code of 404.

If the message contains a *Content-Length* header with a value greater than zero, the server MUST reply with an HTTP error response, as specified in section [3.2.5.1](#), using an HTTP status code of 400.

3.2.5.5 HEAD-REQUEST

The server MUST respond with the HTTP headers that would be generated for the corresponding GET request, but with no message body. This follows the recommendations for HEAD requests, as specified in [\[RFC2616\]](#) section 9.4.

3.2.6 Timer Events

No timer events are required.

3.2.7 Other Local Events

No other local events are defined.

4 Protocol Example

The following section describes a successful search with two servers to illustrate the function of the BITS Peer-Caching: Content Retrieval Protocol.

4.1 Successful Search with Two Servers

This example shows a client searching for the URL
"http://au.download.windowsupdate.com/msdownload/update/v3-19990518/cabpool/mpas-
fe_424732ca30169e03f76401cec04764f02cc6bc3f.exe" in an environment with two servers,
"jroberts19" and "jroberts17".

The client first searches for the URL of interest. It opens a connection to each server. The
[DISCOVERY-REQUEST](#) to "jroberts19" contains the following HTTP header fields and message body:

```
0000 50 4f 53 54 20 2f 42 49-54 53 2d 70 65 65 72 2d POST /BITS-peer-
0010 63 61 63 68 69 6e 67 20-48 54 54 50 2f 31 2e 31 caching HTTP/1.1
0020 0d 0a 41 63 63 65 70 74-3a 20 2a 2f 2a 0d 0a 58 ..Accept: /*.*X
0030 2d 45 54 57 2d 41 43 54-49 56 49 54 59 2d 49 44 -ETW-ACTIVITY-ID
0040 3a 20 7b 41 35 43 34 31-34 43 36 2d 39 34 41 43 : {A5C414C6-94AC
0050 2d 34 33 31 39 2d 38 45-38 44 2d 34 43 33 30 30 -4319-8E8D-4C300
0060 39 31 35 43 44 39 42 7d-0d 0a 55 73 65 72 2d 41 915CD9B)..User-A
0070 67 65 6e 74 3a 20 42 49-54 53 0d 0a 48 6f 73 74 gent: BITS..Host
0080 3a 20 6a 72 6f 62 65 72-74 73 31 39 2e 6e 74 64 : jroberts19.ntd
0090 65 76 2e 63 6f 72 70 2e-6d 69 63 72 6f 73 6f 66 ev.corp.microsof
00A0 74 2e 63 6f 6d 3a 32 31-37 38 0d 0a 43 6f 6e 74 t.com:2178..Cont
00B0 65 6e 74 2d 4c 65 6e 67-74 68 3a 20 36 39 30 0d ent-Length: 690.
00C0 0a 43 6f 6e 6e 65 63 74-69 6f 6e 3a 20 4b 65 65 .Connection: Kee
00D0 70 2d 41 6c 69 76 65 0d-0a 0d 0a 3c 00 3f 00 78 p-Alive....<?.x
00E0 00 6d 00 6c 00 20 00 76-00 65 00 72 00 73 00 69 .m.l. .v.e.r.s.i
00F0 00 6f 00 6e 00 3d 00 22-00 31 00 2e 00 30 00 22 .o.n.=."1...0."
0100 00 20 00 65 00 6e 00 63-00 6f 00 64 00 69 00 6e . .e.n.C.o.d.i.n
0110 00 67 00 3d 00 22 00 75-00 74 00 66 00 2d 00 31 .g.="u.t.f.-1
0120 00 36 00 22 00 3f 00 3e-00 0d 00 0a 00 3c 00 53 .6."?>.....<.S
0130 00 65 00 61 00 72 00 63-00 68 00 52 00 65 00 71 .e.a.r.c.h.R.e.q
0140 00 75 00 65 00 73 00 74-00 3e 00 0d 00 0a 00 20 .u.e.s.t.>.....
0150 00 20 00 20 00 20 00 3c-00 4f 00 72 00 69 00 67 . . .<.O.r.i.g
0160 00 69 00 6e 00 55 00 72-00 6c 00 3e 00 22 00 68 .i.n.U.r.l.>."h
0170 00 74 00 74 00 70 00 3a-00 2f 00 2f 00 61 00 75 .t.t.p.:././a.u
0180 00 2e 00 64 00 6f 00 77-00 6e 00 6c 00 6f 00 61 ...d.o.w.n.l.o.a
0190 00 64 00 2e 00 77 00 69-00 6e 00 64 00 6f 00 77 .d...w.i.n.d.o.w
01A0 00 73 00 75 00 70 00 64-00 61 00 74 00 65 00 2e .s.u.p.d.a.t.e..
01B0 00 63 00 6f 00 6d 00 2f-00 6d 00 73 00 64 00 6f .c.o.m./m.s.d.o
01C0 00 77 00 6e 00 6c 00 6f-00 61 00 64 00 2f 00 75 .w.n.l.o.a.d./u
01D0 00 70 00 64 00 61 00 74-00 65 00 2f 00 76 00 33 .p.d.a.t.e./v.3
01E0 00 2d 00 31 00 39 00 39-00 39 00 30 00 35 00 31 .-1.9.9.9.0.5.1
01F0 00 38 00 2f 00 63 00 61-00 62 00 70 00 6f 00 6f .8./c.a.b.p.o.o
0200 00 6c 00 2f 00 6d 00 70-00 61 00 73 00 2d 00 66 .l./m.p.a.s.-f
0210 00 65 00 5f 00 34 00 32-00 34 00 37 00 33 00 32 .e._4.2.4.7.3.2
0220 00 63 00 61 00 33 00 30-00 31 00 36 00 39 00 65 .c.a.3.0.1.6.9.e
0230 00 30 00 33 00 66 00 37-00 36 00 34 00 30 00 31 .0.3.f.7.6.4.0.1
0240 00 63 00 65 00 63 00 30-00 34 00 37 00 36 00 34 .c.e.c.0.4.7.6.4
0250 00 66 00 30 00 32 00 63-00 63 00 36 00 62 00 63 .f.0.2.c.c.6.b.c
0260 00 33 00 66 00 2e 00 65-00 78 00 65 00 22 00 3c .3.f...e.x.e."<
0270 00 2f 00 4f 00 72 00 69-00 67 00 69 00 6e 00 55 ./O.r.i.g.i.n.U
0280 00 72 00 6c 00 3e 00 0d-00 0a 00 20 00 20 00 20 .r.l.>.....
0290 00 20 00 3c 00 46 00 69-00 6c 00 65 00 4d 00 6f . .<.F.i.l.l.e.M.o
02A0 00 64 00 69 00 66 00 69-00 63 00 61 00 74 00 69 .d.i.f.f.i.c.a.t.i
02B0 00 6f 00 6e 00 54 00 69-00 6d 00 65 00 3e 00 22 .o.n.T.i.m.e.>."
02C0 00 32 00 30 00 30 00 36-00 2d 00 31 00 31 00 2d .2.0.0.6.-1.1.-
02D0 00 30 00 37 00 54 00 31-00 38 00 3a 00 32 00 31 .0.7.T.1.8.:2.1
```

02E0	00 3a 00 34 00 31 00 2e-00 30 00 30 00 30 00 5a	..4.1...0.0.0.Z
02F0	00 22 00 3c 00 2f 00 46-00 69 00 6c 00 65 00 4d	".<./F.i.l.l.e.M
0300	00 6f 00 64 00 69 00 66-00 69 00 63 00 61 00 74	.o.d.i.f.i.c.a.t
0310	00 69 00 6f 00 6e 00 54-00 69 00 6d 00 65 00 3e	.i.o.n.T.i.m.e.>
0320	00 0d 00 0a 00 20 00 20-00 20 00 20 00 3c 00 4d<.M
0330	00 61 00 78 00 52 00 65-00 63 00 6f 00 72 00 64	.a.x.R.e.c.o.r.d
0340	00 73 00 3e 00 22 00 35-00 22 00 3c 00 2f 00 4d	.s.>."5."<./M
0350	00 61 00 78 00 52 00 65-00 63 00 6f 00 72 00 64	.a.x.R.e.c.o.r.d
0360	00 73 00 3e 00 0d 00 0a-00 3c 00 2f 00 53 00 65	.s.>.....<./S.e
0370	00 61 00 72 00 63 00 68-00 52 00 65 00 71 00 75	.a.r.c.h.R.e.q.u
0380	00 65 00 73 00 74 00 3e-00 0d 00 0a 00	.e.s.t.>.....

The request sent to "jroberts17" is similar:

0000	50 4f 53 54 20 2f 42 49-54 53 2d 70 65 65 72 2d	POST /BITS-peer-
0010	63 61 63 68 69 6e 67 20-48 54 54 50 2f 31 2e 31	caching HTTP/1.1
0020	0d 0a 41 63 63 65 70 74-3a 20 2a 2f 2a 0d 0a 58	..Accept: */*.X
0030	2d 45 54 57 2d 41 43 54-49 56 49 54 59 2d 49 44	-ETW-ACTIVITY-ID
0040	3a 20 7b 34 34 30 30 33-45 42 36 2d 43 30 36 35	: {44003EB6-C065
0050	2d 34 39 35 31 2d 41 45-31 38 2d 44 41 31 41 38	-4951-AE18-DA1A8
0060	42 36 43 31 35 32 44 7d-0d 0a 55 73 65 72 2d 41	B6C152D)..User-A
0070	67 65 6e 74 3a 20 42 49-54 53 0d 0a 48 6f 73 74	gent: BITS..Host
0080	3a 20 6a 72 6f 62 65 72-74 73 31 37 2e 6e 74 64	: jroberts17.ntd
0090	65 76 2e 63 6f 72 70 2e-6d 69 63 72 6f 73 6f 66	ev.corp.microsof
00A0	74 2e 63 6f 6d 3a 32 31-37 38 0d 0a 43 6f 6e 74	t.com:2178..Cont
00B0	65 6e 74 2d 4c 65 6e 67-74 68 3a 20 36 39 30 0d	ent-Length: 690.
00C0	0a 43 6f 6e 6e 65 63 74-69 6f 6e 3a 20 4b 65 65	.Connection: Kee
00D0	70 2d 41 6c 69 76 65 0d-0a 0d 0a 3c 00 3f 00 78	p-Alive....<?.x
00E0	00 6d 00 6c 00 20 00 76-00 65 00 72 00 73 00 69	.m.l. .v.e.r.s.i
00F0	00 6f 00 6e 00 3d 00 22-00 31 00 2e 00 30 00 22	.o.n="."1...0."
0100	00 20 00 65 00 6e 00 63-00 6f 00 64 00 69 00 6e	. .e.n.c.o.d.i.n
0110	00 67 00 3d 00 22 00 75-00 74 00 66 00 2d 00 31	.g="."u.t.f.-1
0120	00 36 00 22 00 3f 00 3e-00 0d 00 0a 00 3c 00 53	.6."?>.....<S
0130	00 65 00 61 00 72 00 63-00 68 00 52 00 65 00 71	.e.a.r.c.h.R.e.q
0140	00 75 00 65 00 73 00 74-00 3e 00 0d 00 0a 00 20	.u.e.s.t.>.....
0150	00 20 00 20 00 20 00 3c-00 4f 00 72 00 69 00 67	. . .<.O.r.i.g
0160	00 69 00 6e 00 55 00 72-00 6c 00 3e 00 22 00 68	.i.n.U.r.l.>."h
0170	00 74 00 74 00 70 00 3a-00 2f 00 2f 00 61 00 75	.t.t.p.:././a.u
0180	00 2e 00 64 00 6f 00 77-00 6e 00 6c 00 6f 00 61	...d.o.w.n.l.o.a
0190	00 64 00 2e 00 77 00 69-00 6e 00 64 00 6f 00 77	.d...w.i.n.d.o.w
01A0	00 73 00 75 00 70 00 64-00 61 00 74 00 65 00 2e	.s.u.p.d.a.t.e..
01B0	00 63 00 6f 00 6d 00 2f-00 6d 00 73 00 64 00 6f	.c.o.m./m.s.d.o
01C0	00 77 00 6e 00 6c 00 6f-00 61 00 64 00 2f 00 75	.w.n.l.o.a.d./u
01D0	00 70 00 64 00 61 00 74-00 65 00 2f 00 76 00 33	.p.d.a.t.e./v.3
01E0	00 2d 00 31 00 39 00 39-00 39 00 30 00 35 00 31	-.1.9.9.9.0.5.1
01F0	00 38 00 2f 00 63 00 61-00 62 00 70 00 6f 00 6f	.8./c.a.b.p.o.o
0200	00 6c 00 2f 00 6d 00 70-00 61 00 73 00 2d 00 66	.l./m.p.a.s.-f
0210	00 65 00 5f 00 34 00 32-00 34 00 37 00 33 00 32	.e..4.2.4.7.3.2
0220	00 63 00 61 00 33 00 30-00 31 00 36 00 39 00 65	.c.a.3.0.1.6.9.e
0230	00 30 00 33 00 66 00 37-00 36 00 34 00 30 00 31	.0.3.f.7.6.4.0.1
0240	00 63 00 65 00 63 00 30-00 34 00 37 00 36 00 34	.c.e.c.0.4.7.6.4
0250	00 66 00 30 00 32 00 63-00 63 00 36 00 62 00 63	.f.0.2.c.c.6.b.c
0260	00 33 00 66 00 2e 00 65-00 78 00 65 00 22 00 3c	.3.f...e.x.e."<
0270	00 2f 00 4f 00 72 00 69-00 67 00 69 00 6e 00 55	./O.r.i.g.i.n.U
0280	00 72 00 6c 00 3e 00 0d-00 0a 00 20 00 20 00 20	.r.l.>.....
0290	00 20 00 3c 00 46 00 69-00 6c 00 65 00 4d 00 6f	. .<.F.i.l.l.e.M.o
02A0	00 64 00 69 00 66 00 69-00 63 00 61 00 74 00 69	.d.i.f.i.c.a.t.i
02B0	00 6f 00 6e 00 54 00 69-00 6d 00 65 00 3e 00 22	.o.n.T.i.m.e.>."
02C0	00 32 00 30 00 30 00 36-00 2d 00 31 00 31 00 2d	.2.0.0.6.-.1.1.-
02D0	00 30 00 37 00 54 00 31-00 38 00 3a 00 32 00 31	.0.7.T.1.8.:.2.1
02E0	00 3a 00 34 00 31 00 2e-00 30 00 30 00 30 00 5a	..4.1...0.0.0.Z

02F0	00 22 00 3c 00 2f 00 46-00 69 00 6c 00 65 00 4d	."<./F.i.l.l.e.M
0300	00 6f 00 64 00 69 00 66-00 69 00 63 00 61 00 74	.o.d.i.f.i.c.a.t
0310	00 69 00 6f 00 6e 00 54-00 69 00 6d 00 65 00 3e	.i.o.n.T.i.m.e.>
0320	00 0d 00 0a 00 20 00 20-00 20 00 20 00 3c 00 4d<.M
0330	00 61 00 78 00 52 00 65-00 63 00 6f 00 72 00 64	.a.x.R.e.c.o.r.d
0340	00 73 00 3e 00 22 00 35-00 22 00 3c 00 2f 00 4d	.s.>".5."<./M
0350	00 61 00 78 00 52 00 65-00 63 00 6f 00 72 00 64	.a.x.R.e.c.o.r.d
0360	00 73 00 3e 00 0d 00 0a-00 3c 00 2f 00 53 00 65	.s.>.....<./S.e
0370	00 61 00 72 00 63 00 68-00 52 00 65 00 71 00 75	.a.r.c.h.R.e.q.u
0380	00 65 00 73 00 74 00 3e-00 0d 00 0a 00	.e.s.t.>.....

Server "jroberts19" holds cached data, and responds with an affirmative [DISCOVERY-RESPONSE](#):

0000	48 54 54 50 2f 31 2e 31-20 32 30 30 20 0d 0a 43	HTTP/1.1 200 ..C
0010	6f 6e 74 65 6e 74 2d 4c-65 6e 67 74 68 3a 20 32	ontent-Length: 2
0020	30 35 30 0d 0a 43 6f 6e-74 65 6e 74 2d 54 79 70	050..Content-Typ
0030	65 3a 20 61 70 70 6c 69-63 61 74 69 6f 6e 2f 6f	e: application/o
0040	63 74 65 74 2d 73 74 72-65 61 6d 0d 0a 53 65 72	ctet-stream..Ser
0050	76 65 72 3a 20 4d 69 63-72 6f 73 6f 66 74 2d 48	ver: Microsoft-H
0060	54 54 50 41 50 49 2f 32-2e 30 0d 0a 44 61 74 65	TTPAPI/2.0..Date
0070	3a 20 54 68 75 2c 20 30-39 20 4e 6f 76 20 32 30	: Thu, 09 Nov 20
0080	30 36 20 32 30 3a 35 35-3a 33 37 20 47 4d 54 0d	06 20:55:37 GMT.
0090	0a 0d 0a 3c 00 3f 00 78-00 6d 00 6c 00 20 00 76	...<?.x.m.l. .v
00A0	00 65 00 72 00 73 00 69-00 6f 00 6e 00 3d 00 22	.e.r.s.i.o.n.=."
00B0	00 31 00 2e 00 30 00 22-00 20 00 65 00 6e 00 63	.1...0". .e.n.c
00C0	00 6f 00 64 00 69 00 6e-00 67 00 3d 00 22 00 75	.o.d.i.n.g.=."u
00D0	00 74 00 66 00 2d 00 31-00 36 00 22 00 3f 00 3e	.t.f.-.1.6."?.>
00E0	00 0d 00 0a 00 3c 00 53-00 65 00 61 00 72 00 63<.S.e.a.r.c
00F0	00 68 00 52 00 65 00 73-00 75 00 6c 00 74 00 73	.h.R.e.s.u.l.t.s
0100	00 3e 00 0d 00 0a 00 20-00 20 00 20 00 20 00 3c	.>.....<
0110	00 53 00 74 00 61 00 74-00 75 00 73 00 3e 00 22	.S.t.a.t.u.s.>."
0120	00 53 00 75 00 63 00 63-00 65 00 73 00 73 00 22	.S.u.c.c.e.s.s."
0130	00 3c 00 2f 00 53 00 74-00 61 00 74 00 75 00 73	<./S.t.a.t.u.s
0140	00 3e 00 0d 00 0a 00 20-00 20 00 20 00 20 00 3c	.>.....<
0150	00 43 00 61 00 63 00 68-00 65 00 52 00 65 00 63	.C.a.c.h.e.R.e.c
0160	00 6f 00 72 00 64 00 3e-00 0d 00 0a 00 20 00 20	.o.r.d.>.....
0170	00 20 00 20 00 20 00 20-00 20 00 20 00 3c 00 49<.I
0180	00 64 00 3e 00 22 00 7b-00 36 00 45 00 31 00 42	.d>".{.6.E.l.B
0190	00 30 00 39 00 45 00 46-00 2d 00 39 00 35 00 34	.0.9.E.F.-.9.5.4
01A0	00 46 00 2d 00 34 00 45-00 43 00 32 00 2d 00 42	.F.-.4.E.C.2.-.B
01B0	00 43 00 44 00 42 00 2d-00 30 00 41 00 30 00 46	.C.D.B.-.0.A.0.F
01C0	00 31 00 41 00 34 00 43-00 39 00 31 00 43 00 34	.1.A.4.C.9.l.C.4
01D0	00 7d 00 22 00 3c 00 2f-00 49 00 64 00 3e 00 0d	.}."<./I.d.>..
01E0	00 0a 00 20 00 20 00 20-00 20 00 20 00 20 00 20
01F0	00 20 00 3c 00 43 00 72-00 65 00 61 00 74 00 69	.<.C.r.e.a.t.i
0200	00 6f 00 6e 00 54 00 69-00 6d 00 65 00 3e 00 22	.o.n.T.i.m.e>."
0210	00 32 00 30 00 30 00 36-00 2d 00 31 00 31 00 2d	.2.0.0.6.-.1.1.-
0220	00 30 00 39 00 54 00 32-00 30 00 3a 00 35 00 34	.0.9.T.2.0.:.5.4
0230	00 3a 00 34 00 37 00 2e-00 34 00 33 00 37 00 5a	..4.7...4.3.7.Z
0240	00 22 00 3c 00 2f 00 43-00 72 00 65 00 61 00 74	."<./C.r.e.a.t
0250	00 69 00 6f 00 6e 00 54-00 69 00 6d 00 65 00 3e	.i.o.n.T.i.m.e>
0260	00 0d 00 0a 00 20 00 20-00 20 00 20 00 20 00 20
0270	00 20 00 20 00 3c 00 4d-00 6f 00 64 00 69 00 66	...<.M.o.d.i.f
0280	00 69 00 63 00 61 00 74-00 69 00 6f 00 6e 00 54	.i.c.a.t.i.o.n.T
0290	00 69 00 6d 00 65 00 3e-00 22 00 32 00 30 00 30	.i.m.e>".2.0.0
02A0	00 36 00 2d 00 31 00 31-00 2d 00 30 00 39 00 54	.6.-.1.1.-.0.9.T
02B0	00 32 00 30 00 3a 00 35-00 34 00 3a 00 35 00 38	.2.0.:.5.4.:.5.8
02C0	00 2e 00 36 00 30 00 37-00 5a 00 22 00 3c 00 2f	...6.0.7.Z."<./
02D0	00 4d 00 6f 00 64 00 69-00 66 00 69 00 63 00 61	.M.o.d.i.f.i.c.a
02E0	00 74 00 69 00 6f 00 6e-00 54 00 69 00 6d 00 65	.t.i.o.n.T.i.m.e
02F0	00 3e 00 0d 00 0a 00 20-00 20 00 20 00 20 00 20	.>.....
0300	00 20 00 20 00 20 00 3c-00 4c 00 61 00 73 00 74	...<.L.a.s.t

0310	00	41	00	63	00	63	00	65-00	73	00	73	00	54	00	69	.A.c.c.e.s.s.T.i
0320	00	6d	00	65	00	3e	00	22-00	32	00	30	00	30	00	36	.m.e.>".2.0.0.6
0330	00	2d	00	31	00	31	00	2d-00	30	00	39	00	54	00	32	.-.1.1.-.0.9.T.2
0340	00	30	00	3a	00	35	00	34-00	3a	00	35	00	38	00	2e	.0.:.5.4.:.5.8..
0350	00	36	00	30	00	37	00	5a-00	22	00	3c	00	2f	00	4c	.6.0.7.Z."<./L
0360	00	61	00	73	00	74	00	41-00	63	00	63	00	65	00	73	.a.s.t.A.c.c.e.s
0370	00	73	00	54	00	69	00	6d-00	65	00	3e	00	0d	00	0a	.s.T.i.m.e>....
0380	00	20	00	20	00	20	00	20-00	20	00	20	00	20	00	20
0390	00	3c	00	4f	00	72	00	69-00	67	00	69	00	6e	00	55	<.O.r.i.g.i.n.U
03A0	00	72	00	6c	00	3e	00	22-00	68	00	74	00	74	00	70	.r.l.>".h.t.t.p
03B0	00	3a	00	2f	00	2f	00	61-00	75	00	2e	00	64	00	6f/.a.u...d.o
03C0	00	77	00	6e	00	6c	00	6f-00	61	00	64	00	2e	00	77	.w.n.l.o.a.d...w
03D0	00	69	00	6e	00	64	00	6f-00	77	00	73	00	75	00	70	.i.n.d.o.w.s.u.p
03E0	00	64	00	61	00	74	00	65-00	2e	00	63	00	6f	00	6d	.d.a.t.e...c.o.m
03F0	00	2f	00	6d	00	73	00	64-00	6f	00	77	00	6e	00	6c	./m.s.d.o.w.n.l
0400	00	6f	00	61	00	64	00	2f-00	75	00	70	00	64	00	61	.o.a.d./u.p.d.a
0410	00	74	00	65	00	2f	00	76-00	33	00	2d	00	31	00	39	.t.e./v.3.-.1.9
0420	00	39	00	39	00	30	00	35-00	31	00	38	00	2f	00	63	.9.9.0.5.1.8./c
0430	00	61	00	62	00	70	00	6f-00	6f	00	6c	00	2f	00	6d	.a.b.p.o.o.l./m
0440	00	70	00	61	00	73	00	2d-00	66	00	65	00	5f	00	34	.p.a.s.-f.e._.4
0450	00	32	00	34	00	37	00	33-00	32	00	63	00	61	00	33	.2.4.7.3.2.c.a.3
0460	00	30	00	31	00	36	00	39-00	65	00	30	00	33	00	66	.0.1.6.9.e.0.3.f
0470	00	37	00	36	00	34	00	30-00	31	00	63	00	65	00	63	.7.6.4.0.1.c.e.c
0480	00	30	00	34	00	37	00	36-00	34	00	66	00	30	00	32	.0.4.7.6.4.f.0.2
0490	00	63	00	63	00	36	00	62-00	63	00	33	00	66	00	2e	.c.c.6.b.c.3.f..
04A0	00	65	00	78	00	65	00	22-00	3c	00	2f	00	4f	00	72	.e.x.e."<./O.r
04B0	00	69	00	67	00	69	00	6e-00	55	00	72	00	6c	00	3e	.i.g.i.n.U.r.l.>
04C0	00	0d	00	0a	00	20	00	20-00	20	00	20	00	20	00	20
04D0	00	20	00	20	00	3c	00	4c-00	6f	00	63	00	61	00	6c	. . .<.L.o.c.a.l
04E0	00	55	00	72	00	6c	00	3e-00	22	00	42	00	49	00	54	.U.r.l.>".B.I.T
04F0	00	53	00	2d	00	70	00	65-00	65	00	72	00	2d	00	63	.S.-.p.e.e.r.-c
0500	00	61	00	63	00	68	00	69-00	6e	00	67	00	2f	00	7b	.a.c.h.i.n.g./.{
0510	00	36	00	45	00	31	00	42-00	30	00	39	00	45	00	46	.6.E.1.B.0.9.E.F
0520	00	2d	00	39	00	35	00	34-00	46	00	2d	00	34	00	45	.-.9.5.4.F.-.4.E
0530	00	43	00	32	00	2d	00	42-00	43	00	44	00	42	00	2d	.C.2.-.B.C.D.B.-
0540	00	30	00	41	00	30	00	46-00	31	00	41	00	34	00	43	.0.A.0.F.1.A.4.C
0550	00	39	00	31	00	43	00	34-00	7d	00	22	00	3c	00	2f	.9.1.C.4.}>".<./
0560	00	4c	00	6f	00	63	00	61-00	6c	00	55	00	72	00	6c	.L.o.c.a.l.U.r.l
0570	00	3e	00	0d	00	0a	00	20-00	20	00	20	00	20	00	20	>.....
0580	00	20	00	20	00	20	00	3c-00	46	00	69	00	6c	00	65	. . .<.F.i.l.e
0590	00	4d	00	6f	00	64	00	69-00	66	00	69	00	63	00	61	.M.o.d.i.f.i.c.a
05A0	00	74	00	69	00	6f	00	6e-00	54	00	69	00	6d	00	65	.t.i.o.n.T.i.m.e
05B0	00	3e	00	22	00	32	00	30-00	30	00	36	00	2d	00	31	>".2.0.0.6.-.1
05C0	00	31	00	2d	00	30	00	37-00	54	00	31	00	38	00	3a	.1.-.0.7.T.1.8.:
05D0	00	32	00	31	00	3a	00	34-00	31	00	2e	00	30	00	30	.2.1.:.4.1...0.0
05E0	00	30	00	5a	00	22	00	3c-00	2f	00	46	00	69	00	6c	.0.Z."<./F.i.l
05F0	00	65	00	4d	00	6f	00	64-00	69	00	66	00	69	00	63	.e.M.o.d.i.f.i.c
0600	00	61	00	74	00	69	00	6f-00	6e	00	54	00	69	00	6d	.a.t.i.o.n.T.i.m
0610	00	65	00	3e	00	0d	00	0a-00	20	00	20	00	20	00	20	.e>.....
0620	00	20	00	20	00	20	00	20-00	3c	00	46	00	69	00	6c<.F.i.l
0630	00	65	00	53	00	69	00	7a-00	65	00	3e	00	22	00	33	.e.S.i.z.e>".3
0640	00	33	00	37	00	33	00	33-00	38	00	34	00	22	00	3c	.3.7.3.3.8.4."<
0650	00	2f	00	46	00	69	00	6c-00	65	00	53	00	69	00	7a	./F.i.l.l.e.S.i.z
0660	00	65	00	3e	00	0d	00	0a-00	20	00	20	00	20	00	20	.e>.....
0670	00	20	00	20	00	20	00	20-00	3c	00	43	00	6f	00	6e<.C.o.n
0680	00	74	00	65	00	6e	00	74-00	52	00	61	00	6e	00	67	.t.e.n.t.R.a.n.g
0690	00	65	00	3e	00	0d	00	0a-00	20	00	20	00	20	00	20	.e>.....
06A0	00	20	00	20	00	20	00	20-00	20	00	20	00	20	00	20
06B0	00	3c	00	4f	00	66	00	66-00	73	00	65	00	74	00	3e	<.O.f.f.s.e.t.>
06C0	00	22	00	31	00	30	00	30-00	22	00	3c	00	2f	00	4f	."1.0.0."<./O
06D0	00	66	00	66	00	73	00	65-00	74	00	3e	00	0d	00	0a	.f.f.s.e.t>....
06E0	00	20	00	20	00	20	00	20-00	20	00	20	00	20	00	20
06F0	00	20	00	20	00	20	00	20-00	3c	00	4c	00	65	00	6e<.L.e.n
0700	00	67	00	74	00	68	00	3e-00	22	00	31	00	36	00	22	.g.t.h.>".1.6."
0710	00	3c	00	2f	00	4c	00	65-00	6e	00	67	00	74	00	68	<./L.e.n.g.t.h

```

0720 00 3e 00 0d 00 0a 00 20-00 20 00 20 00 20 00 20 .>.....
0730 00 20 00 20 00 20 00 3c-00 2f 00 43 00 6f 00 6e . . . .<./C.o.n
0740 00 74 00 65 00 6e 00 74-00 52 00 61 00 6e 00 67 .t.e.n.t.R.a.n.g
0750 00 65 00 3e 00 0d 00 0a-00 20 00 20 00 20 00 20 .e.>.....
0760 00 20 00 20 00 20 00 20-00 3c 00 43 00 6f 00 6e . . . .<./C.o.n
0770 00 74 00 65 00 6e 00 74-00 52 00 61 00 6e 00 67 .t.e.n.t.R.a.n.g
0780 00 65 00 3e 00 0d 00 0a-00 20 00 20 00 20 00 20 .e.>.....
0790 00 20 00 20 00 20 00 20-00 20 00 20 00 20 00 20 . . . .<./C.o.n
07A0 00 3c 00 4f 00 66 00 66-00 73 00 65 00 74 00 3e .<./O.f.f.s.e.t.>
07B0 00 22 00 32 00 30 00 30-00 22 00 3c 00 2f 00 4f .".2.0.0."<./O
07C0 00 66 00 66 00 73 00 65-00 74 00 3e 00 0d 00 0a .f.f.s.e.t.>....
07D0 00 20 00 20 00 20 00 20-00 20 00 20 00 20 00 20 . . . .<./C.o.n
07E0 00 20 00 20 00 20 00 20-00 3c 00 4c 00 65 00 6e . . . .<./L.e.n
07F0 00 67 00 74 00 68 00 3e-00 22 00 34 00 38 00 22 .g.t.h.>".4.8."
0800 00 3c 00 2f 00 4c 00 65-00 6e 00 67 00 74 00 68 .<./L.e.n.g.t.h
0810 00 3e 00 0d 00 0a 00 20-00 20 00 20 00 20 00 20 .>.....
0820 00 20 00 20 00 20 00 3c-00 2f 00 43 00 6f 00 6e . . . .<./C.o.n
0830 00 74 00 65 00 6e 00 74-00 52 00 61 00 6e 00 67 .t.e.n.t.R.a.n.g
0840 00 65 00 3e 00 0d 00 0a-00 20 00 20 00 20 00 20 .e.>.....
0850 00 3c 00 2f 00 43 00 61-00 63 00 68 00 65 00 52 .<./C.a.c.h.e.R
0860 00 65 00 63 00 6f 00 72-00 64 00 3e 00 0d 00 0a .e.c.o.r.d.>....
0870 00 3c 00 2f 00 53 00 65-00 61 00 72 00 63 00 68 .<./S.e.a.r.c.h
0880 00 52 00 65 00 73 00 75-00 6c 00 74 00 73 00 3e .R.e.s.u.l.t.s.>
0890 00 0d 00 0a 00 . . . .

```

Server "jroberts17" does not hold cached data, and responds negatively:

```

0000 48 54 54 50 2f 31 2e 31-20 32 30 30 20 0d 0a 43 HTTP/1.1 200 ..C
0010 6f 6e 74 65 6e 74 2d 4c-65 6e 67 74 68 3a 20 32 ontent-Length: 2
0020 33 32 0d 0a 43 6f 6e 74-65 6e 74 2d 54 79 70 65 32..Content-Type
0030 3a 20 61 70 70 6c 69 63-61 74 69 6f 6e 2f 6f 63 : application/oc
0040 74 65 74 2d 73 74 72 65-61 6d 0d 0a 53 65 72 76 tet-stream..Serv
0050 65 72 3a 20 4d 69 63 72-6f 73 6f 66 74 2d 48 54 er: Microsoft-HT
0060 54 50 41 50 49 2f 32 2e-30 0d 0a 44 61 74 65 3a TPAPI/2.0..Date:
0070 20 54 68 75 2c 20 30 39-20 4e 6f 76 20 32 30 30 Thu, 09 Nov 200
0080 36 20 30 32 3a 30 30 3a-33 33 20 47 4d 54 0d 0a 6 02:00:33 GMT..
0090 0d 0a 3c 00 3f 00 78 00-6d 00 6c 00 20 00 76 00 ..<?.x.m.l. .v.
00A0 65 00 72 00 73 00 69 00-6f 00 6e 00 3d 00 22 00 e.r.s.i.o.n.=.".
00B0 31 00 2e 00 30 00 22 00-20 00 65 00 6e 00 63 00 1...0." .e.n.c.
00C0 6f 00 64 00 69 00 6e 00-67 00 3d 00 22 00 75 00 o.d.i.n.g.=.".u.
00D0 74 00 66 00 2d 00 31 00-36 00 22 00 3f 00 3e 00 t.f.-.1.6."?>.
00E0 0d 00 0a 00 3c 00 53 00-65 00 61 00 72 00 63 00 ....<S.e.a.r.c.
00F0 68 00 52 00 65 00 73 00-75 00 6c 00 74 00 73 00 h.R.e.s.u.l.t.s.
0100 3e 00 0d 00 0a 00 20 00-20 00 20 00 20 00 3c 00 >..... . . . .<
0110 53 00 74 00 61 00 74 00-75 00 73 00 3e 00 22 00 S.t.a.t.u.s.>.".
0120 43 00 6f 00 6e 00 74 00-65 00 6e 00 74 00 4e 00 C.o.n.t.e.n.t.N.
0130 6f 00 74 00 46 00 6f 00-75 00 6e 00 64 00 22 00 o.t.F.o.u.n.d.".
0140 3c 00 2f 00 53 00 74 00-61 00 74 00 75 00 73 00 <./S.t.a.t.u.s.
0150 3e 00 0d 00 0a 00 3c 00-2f 00 53 00 65 00 61 00 >.....<./S.e.a.
0160 72 00 63 00 68 00 52 00-65 00 73 00 75 00 6c 00 r.c.h.R.e.s.u.l.
0170 74 00 73 00 3e 00 0d 00-0a 00 t.s.>.....

```

The client requests cached data from "jroberts19" by using a [DOWNLOAD-REQUEST](#):

```

0000 47 45 54 20 2f 42 49 54-53 2d 70 65 65 72 2d 63 GET /BITS-peer-c
0010 61 63 68 69 6e 67 2f 25-37 42 36 45 31 42 30 39 aching/%7B6E1B09
0020 45 46 2d 39 35 34 46 2d-34 45 43 32 2d 42 43 44 EF-954F-4EC2-BCD
0030 42 2d 30 41 30 46 31 41-34 43 39 31 43 34 25 37 B-0A0F1A4C91C4%7

```


0040	44 20 48 54 54 50 2f 31-2e 31 0d 0a 41 63 63 65	D HTTP/1.1..Acce
0050	70 74 3a 20 2a 2f 2a 0d-0a 41 63 63 65 70 74 2d	pt: */*..Accept-
0060	45 6e 63 6f 64 69 6e 67-3a 20 69 64 65 6e 74 69	Encoding: identi
0070	74 79 0d 0a 52 61 6e 67-65 3a 20 62 79 74 65 73	ty..Range: bytes
0080	3d 30 2d 31 35 0d 0a 49-66 2d 55 6e 6d 6f 64 69	=0-15..If-Unmodi
0090	66 69 65 64 2d 53 69 6e-63 65 3a 20 54 75 65 2c	fied-Since: Tue,
00A0	20 30 37 20 4e 6f 76 20-32 30 30 36 20 31 38 3a	07 Nov 2006 18:
00B0	32 31 3a 34 31 20 47 4d-54 0d 0a 55 73 65 72 2d	21:41 GMT..User-
00C0	41 67 65 6e 74 3a 20 4d-69 63 72 6f 73 6f 66 74	Agent: Microsoft
00D0	20 42 49 54 53 2f 37 2e-30 0d 0a 48 6f 73 74 3a	BITS/7.0..Host:
00E0	20 6a 72 6f 62 65 72 74-73 31 39 2e 6e 74 64 65	jroberts19.ntd
00F0	76 2e 63 6f 72 70 2e 6d-69 63 72 6f 73 6f 66 74	ev.corp.microsof
0100	2e 63 6f 6d 3a 32 31 37-38 0d 0a 43 6f 6e 6e 65	t.com:2178..Conne
0110	63 74 69 6f 6e 3a 20 4b-65 65 70 2d 41 6c 69 76	ction: Keep-Aliv
0120	65 0d 0a 0d 0a	e....

The server replies with a [DOWNLOAD-RESPONSE](#):

0000	48 54 54 50 2f 31 2e 31-20 32 30 36 20 0d 0a 43	HTTP/1.1 206 ..C
0010	6f 6e 74 65 6e 74 2d 4c-65 6e 67 74 68 3a 20 31	ontent-Length: 1
0020	36 0d 0a 43 6f 6e 74 65-6e 74 2d 54 79 70 65 3a	6..Content-Type:
0030	20 61 70 70 6c 69 63 61-74 69 6f 6e 2f 6f 63 74	application/oct
0040	65 74 2d 73 74 72 65 61-6d 0d 0a 43 6f 6e 74 65	et-stream..Conte
0050	6e 74 2d 52 61 6e 67 65-3a 20 62 79 74 65 73 20	nt-Range: bytes
0060	30 2d 31 35 2f 36 34 0d-0a 53 65 72 76 65 72 3a	0-15/64..Server:
0070	20 4d 69 63 72 6f 73 6f-66 74 2d 48 54 54 50 41	Microsoft-HTTP
0080	50 49 2f 32 2e 30 0d 0a-42 49 54 53 5f 42 41 53	PI/2.0..BITS_BAS
0090	49 43 5f 49 4e 46 4f 3a-20 30 78 31 43 37 30 32	IC INFO: 0x1C702
00A0	39 39 39 32 33 42 45 38-38 30 2c 30 78 31 43 37	99923BE880,0x1C7
00B0	30 32 39 39 39 32 33 42-45 38 38 30 2c 30 78 31	0299923BE880,0x1
00C0	43 37 30 32 39 39 39 32-33 42 45 38 38 30 2c 30	C70299923BE880,0
00D0	78 31 43 37 30 32 39 39-39 32 33 42 45 38 38 30	x1C70299923BE880
00E0	2c 30 78 32 30 0d 0a 44-61 74 65 3a 20 54 68 75	,0x20..Date: Thu
00F0	2c 20 30 39 20 4e 6f 76-20 32 30 30 36 20 32 30	, 09 Nov 2006 20
0100	3a 35 35 3a 33 39 20 47-4d 54 0d 0a 0d 0a 20 72	:55:39 GMT.... r
0110	75 6e 20 69 6e 20 44 4f-53 20 6d 6f 64 65	un in DOS mode

5 Security

The following sections specify security considerations for implementers of the BITS Peer-Caching: Content Retrieval Protocol.

5.1 Security Considerations for Implementers

There are no security considerations for implementers.

5.2 Index of Security Parameters

No security parameters are defined.

6 Appendix A: Windows Behavior

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

- Windows Vista
- Windows Server 2008

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

[<1> Section 1.5:](#) Windows identifies servers by using the [Background Intelligent Transfer Service \(BITS\) Peer-Caching: Peer Discovery Protocol](#) (for more information, see [\[MS-BPDP\]](#)).

[<2> Section 1.5:](#) Windows requires that a certificate have the following attributes:

- It must be within its period of validity.
- It must be self-signed.
- The issuer/subject name must be a **security identifier (SID)** representing a machine account in the recipient host's Active Directory **domain**.
- Certificates sent by the client must include the "id-kp-clientAuth" **extended key usage (EKU)** specified in [\[RFC3280\]](#) section 4.2.1.13.
- Certificates sent by the server must include the "id-kp-serverAuth" EKU specified in [\[RFC3280\]](#) section 4.2.1.13.

[<3> Section 2.1:](#) Windows verifies that the received certificate is self-signed and is present in the "BITS\Peers" substore of the CERT_SYSTEM_STORE_SERVICES certificate store.

[<4> Section 2.2:](#) Windows does not define any additional HTTP headers.

[<5> Section 2.2.1.5:](#) The Windows peer server never includes this element. The Windows peer client ignores this element.

[<6> Section 2.2.1.6:](#) The Windows client always sends a value of 5.

[<7> Section 2.2.1.6:](#) Windows ignores records beyond MaxRecords.

[<8> Section 2.2.2.2:](#) Windows sends this header in all requests. The GUID is unique among all [DISCOVERY-REQUESTS](#) sent by a particular client.

[<9> Section 2.2.2.3:](#) Windows ignores attributes in XML elements.

[<10> Section 2.2.3.2:](#) Windows ignores attributes in XML elements.

[<11> Section 2.2.3.2:](#) Windows includes one comment in both success and failure responses; the comment contains the HRESULT of the internal operation in the following format:

```
4*SP "<!-- Error 0x" HR " -->" CRLF
```

<12> [Section 2.2.4:](#) The BITS component of Windows Vista uses HTTP ranges to resume an interrupted download, and also uses ranges for rate control when a "background" download is requested.

<13> [Section 2.2.5:](#) Windows returns status 206 when the request contains a *Content-Range* header, even when the range covers the entire record.

<14> [Section 2.2.5:](#) Windows always provides this header.

<15> [Section 2.2.6:](#) The Windows client sends HEAD requests in some circumstances. Specifically, it sends a HEAD prior to downloading when the priority of the client's BITS job is not `bg_job_priority_foreground`, and it omits the HEAD when the priority is `bg_job_priority_foreground`. For an overview of BITS in Windows, see [\[MSDN-BITS\]](#).

<16> [Section 3.1.1.3.1:](#) Windows initializes an integer `AUTH_NEEDED` as follows: If ≥ 30 percent of servers in the `PEER-SERVER-TABLE` are authenticated, set `AUTH_NEEDED=10`; else set `AUTH_NEEDED=5`.

<17> [Section 3.1.1.3.2:](#) Windows uses the following criteria to choose a peer:

1. If (`AUTH_NEEDED > 0`), set `NEW_PEER` to a random server (not chosen yet by this search) with `authenticated == true`. If found, decrement `AUTH_NEEDED` and return the server; otherwise, set `AUTH_NEEDED` to zero and go to step 2.
2. Choose a random server (not chosen yet by this search) with `authenticated == false`. If found, return the server.

<18> [Section 3.1.1.3.5:](#) Windows triggers a peer-discovery request (for more information, see [\[MS-BPDP\]](#) section 3.2.4.2).

<19> [Section 3.1.2.3:](#) Windows uses a value of five minutes.

<20> [Section 3.1.3:](#) Windows uses the [BITS Peer-Caching: Peer Discovery Protocol](#) (for more information, see [\[MS-BPDP\]](#)) to listen for updates to the table of peers. Windows caches the table of peers between invocations.

<21> [Section 3.1.7.1.1:](#) Windows verifies that the received certificate is self-signed and is present in the "BITS\Peers" substore of the `CERT_SYSTEM_STORE_SERVICES` certificate store.

<22> [Section 3.1.7.1.1:](#) Windows attempts to authenticate the server by an exchange of certificates via the [BITS Peer-Caching: Peer Authentication Protocol](#) (for more information, see [\[MS-BPAU\]](#)).

<23> [Section 3.1.7.2.1:](#) Windows verifies that the received certificate is self-signed and is present in the "BITS\Peers" substore of the `CERT_SYSTEM_STORE_SERVICES` certificate store.

<24> [Section 3.1.7.3.1:](#) Windows removes peers from the table after transport errors.

<25> [Section 3.1.7.3.4:](#) Windows attempts to authenticate the client by an exchange of certificates via the [BITS Peer-Caching: Peer Authentication Protocol](#) (for more information, see [\[MS-BPAU\]](#)).

<26> [Section 3.1.7.3.6:](#) Windows attempts to authenticate the server by an exchange of certificates via the [BITS Peer-Caching: Peer Authentication Protocol](#) (for more information, see [\[MS-BPAU\]](#)).

<27> [Section 3.2.4.1:](#) Windows processes all messages that are already received.

[<28> Section 3.2.5.1:](#) Windows follows these rules except in replies triggered by a failure in validation of the client's certificate. Such replies contain an HTTP entity body in Unicode with the following ABNF structure:

```
Quot = %d34
HR = 8*HEXDIG
Body = "<?xml version=" quot "1.0" quot " encoding=" quot "utf-16" quot ">" CRLF
      "<SearchResults>" CRLF
      4*SP "<!-- Error 0x" HR " -->" CRLF
      4*SP "<Status>" quot RESULT quot "</Status>" CRLF
      "</SearchResults>" CRLF
```

The value of the HR rule in the ANBF represents the internal HRESULT value generated by the server operation that failed. The generated RESULT values are listed in the following table.

HRESULT value	Corresponding RESULT string
0x80040005	"CertificateNotFound"
0x8007000E	"OutOfResources"
Any other value	"AccessDenied"

[<29> Section 3.2.5.2:](#) Windows allows version 1.0 as well.

[<30> Section 3.2.5.2:](#) Windows allows only three messages to be processed simultaneously.

[<31> Section 3.2.5.3:](#) Windows verifies that the received certificate is self-signed and is present in the "BITS\Peers" substore of the CERT_SYSTEM_STORE_SERVICES certificate store.

[<32> Section 3.2.5.3:](#) Windows limits the size of the HTTP header fields to 16 kilobytes, and limits the XML body to 1 megabyte.

[<33> Section 3.2.5.3:](#) Windows always includes an embedded XML comment specifying an HRESULT status. If an error occurs during parsing of the client request, the HRESULT value of the error appears here. The HRESULT is not consumed by the Windows client; it was included only for debugging purposes. The ABNF form of a failure reply, including the comment, is the same as in note <28>.

The value of the <Status> element depends on the parsing error as well, and is specified by the following table.

HRESULT value	Corresponding string in <Status> element
x80070005	"AccessDenied"
x8007000E	"OutOfResources"
x80070057 or x80004001	"InvalidSearch"
Any other value	"Unknown"

If parsing was successful, the HRESULT in the comment block is zero.

[<34> Section 3.2.5.4:](#) Windows verifies that the received certificate is self-signed and is present in the "BITS\Peers" substore of the CERT_SYSTEM_STORE_SERVICES certificate store.

7 Appendix B: XML Schema

The XSD for XML namespace

<http://schemas.microsoft.com/windows/2007/01/BITS/ContentDiscovery> is reproduced as follows:

[XML]

```
<?xml version="1.0"?>
<schema
  targetNamespace="http://schemas.microsoft.com/windows/2007/01/BITS/ContentDiscovery"
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:cd="http://schemas.microsoft.com/windows/2007/01/BITS/ContentDiscovery"
  elementFormDefault="qualified">

  <!-- request types -->

  <simpleType name="guid">
    <restriction base="string">
      <pattern value="[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}" />
    </restriction>
  </simpleType>

  <simpleType name="url">
    <restriction base="string">
      <maxLength value="2200" />
    </restriction>
  </simpleType>

  <!-- request format -->
  <complexType name="searchRequest">
    <sequence>
      <element name="OriginUrl" type="cd:url"/>
      <element name="FileModificationTime" type="dateTime" />
      <element name="FileSize" type="unsignedLong" minOccurs="0"/>
      <element name="FileEtag" type="string" minOccurs="0" />

      <element name="MaxRecords" type="positiveInteger" minOccurs="0" default="1" />

      <any minOccurs="0" maxOccurs="unbounded" processContents="lax"
        namespace="##other"/>
    </sequence>
  </complexType>

  <!-- response types -->

  <simpleType name="searchStatus">
    <restriction base="string">
      <enumeration value="Success"/>
      <enumeration value="CertificateNotFound"/>
      <enumeration value="ContentNotFound"/>
      <enumeration value="AccessDenied"/>
      <enumeration value="OutOfResources"/>
      <enumeration value="InvalidSearch"/>
    </restriction>
  </simpleType>

  <complexType name="fileRange">
    <sequence>
      <element name="Offset" type="unsignedLong"/>
      <element name="Length" type="unsignedLong"/>
    </sequence>
  </complexType>
```

```

<complexType name="cacheRecord">
  <sequence>
    <element name="Id" type="cd:guid"/>
    <element name="CreationTime" type="dateTime"/>
    <element name="ModificationTime" type="dateTime"/>
    <element name="LastAccessTime" type="dateTime"/>
    <element name="OriginUrl" type="cd:url"/>
    <element name="LocalUrl" type="cd:url"/>
    <element name="FileModificationTime" type="dateTime"/>
    <element name="FileSize" type="unsignedLong"/>
    <element name="FileEtag" type="string" minOccurs="0" />
    <element name="ContentRange" type="cd:fileRange" maxOccurs="unbounded"/>
    <any minOccurs="0" maxOccurs="unbounded" processContents="lax"
      namespace="##other"/>
  </sequence>
</complexType>

<!-- response format -->

<complexType name="searchResponse">
  <sequence>
    <element name="Status" type="cd:searchStatus" />
    <element name="CacheRecord" type="cd:cacheRecord" minOccurs="0"
      maxOccurs="unbounded"/>
    <any minOccurs="0" maxOccurs="unbounded" processContents="lax"
      namespace="##other"/>
  </sequence>
</complexType>

  <element name="SearchRequest" type="cd:searchRequest"/>
  <element name="SearchResults" type="cd:searchResponse"/>
</schema>

```

8 Index

A

Abstract data model
 [client](#)
 [server](#)
[Address - new](#)
[Applicability](#)

B

[Body data](#)

C

[cacheRecord](#)
[Call](#)
 [connection failure](#)
 [pending completes](#)
 [server certificate problems](#)
[Call events](#)
[Call request timeout](#) ([section 3.1.2.2](#), [section 3.1.2.3](#))
[Call response](#)
[Call response timeout](#)
[Cancel search](#)
[Capability negotiation](#)
Client
 [abstract data model](#)
 [call events](#)
 [download events](#)
 [higher-layer triggered events](#)
 [initialization](#)
 [local events](#)
 [message processing](#)
 [overview](#)
 [search events](#)
 [sequencing rules](#)
 [timer events](#)
 [timers](#)
[Common data types](#)
[Connection failure during Call](#)
[Connection failure during download](#)

D

Data model - abstract
 [client](#)
 [server](#)
[Data types](#)
[DISCOVERY-REQUEST](#) ([section 2.2.2](#), [section 3.2.5.3](#))
[DISCOVERY-RESPONSE](#)
Download
 [connection failure](#)
 [server certificate problems](#)
[Download events](#)
[Download request](#)
[Download response](#)
[Download response timeout](#)
[DOWNLOAD-REQUEST](#) ([section 2.2.4](#), [section 3.2.5.4](#))
[DOWNLOAD-RESPONSE](#)

E

[Error responses - HTTP-level](#)
[ESULT_UNKNOWN](#)
Examples
 [overview](#)
 [successful search with two servers](#)

F

[Fields - vendor-extensible](#)
[fileRange](#)

G

[Glossary](#)
[guid](#)

H

[HEAD-REQUEST](#) ([section 2.2.6](#), [section 3.2.5.5](#))
[HEAD-RESPONSE](#)
Higher-layer triggered events
 [client](#)
 [server](#)
HTTP header fields ([section 2.2.2.1](#), [section 2.2.2.2](#))
[HTTP headers](#)
[HTTP-level error responses](#)

I

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

L

Local events
 [client](#)
 [server](#)

M

[Message body](#)
Message processing
 [client](#)
 [server](#)
Messages
 [data types](#)
 [overview](#)
 [syntax](#)
 [transport](#)
 [validation](#)

N

[New download request](#)
[New search request](#)
[Normative references](#)
[Notification of new server or address](#)

O

[Overview \(synopsis\)](#)

P

[Parameters - security index](#)
[Pending Call completes](#)
[Preconditions](#)
[Prerequisites](#)

R

References
 [informative](#)
 [normative](#)
 [overview](#)
[Relationship to other protocols](#)
[RESULT_ACCESS_DENIED](#)
[RESULT_CLIENT_CERT_UNKNOWN](#)
[RESULT_FOUND](#)
[RESULT_INVALID_SEARCH](#)
[RESULT_NOT_FOUND](#)
[RESULT_OUT_OF_RESOURCES](#)
[RESULT_SERVER_CERT_UNKNOWN](#)
[RESULT_TRANSPORT_ERROR](#)

S

[Schema - XML](#)
[Search](#)
[Search events](#)
[Search timeout](#) ([section 3.1.2.1](#), [section 3.1.6.3](#))
[searchRequest](#)
[searchResponse](#)
[searchStatus](#)
Security
 [implementer considerations](#)
 [overview](#)
 [parameter index](#)
Sequencing rules
 [client](#)
 [server](#)
Server
 [abstract data model](#)
 [higher-layer triggered events](#)
 [initialization](#)
 [local events](#)
 [message processing](#)
 [notification when new](#)
 [overview](#)
 [sequencing rules](#)
 [timer events](#)
 [timers](#)

[Server certificate problems during Call](#)
[Server certificate problems during download](#)
[Shutdown](#) ([section 3.1.7.3.9](#), [section 3.2.4.1](#))
[Standard HTTP header fields](#)
[Standards assignments](#)
[STATE_CHOOSE_SERVER](#)
[STATE_COMPLETE](#)
[STATE_DISCOVER_SERVERS](#)
[STATE_INIT](#)
[STATE_SEND_REQUEST](#)
[STATE_WAIT](#)
[Successful search with two servers example](#)
[Syntax](#)
[Syntax - data types](#)

T

[Table of cache records](#)
[Table of servers](#)
Timeout
 [call request](#)
 [call response](#)
 [download request](#)
 [download response](#)
 [search](#) ([section 3.1.2.1](#), [section 3.1.6.3](#))
Timer events
 [client](#)
 [server](#)
Timers
 [client](#)
 [server](#)
[Transport](#)
Triggered events - higher-layer
 [client](#)
 [server](#)

U

[url](#)

V

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

W

[Windows behavior](#)

X

[XML schema](#)