[MS-BRWSA]:
Common Internet File System (CIFS) Browser Auxiliary Protocol

Revision Summary

Date	Revision History	Revision Class	Comments
09/28/2007	0.1	Major	MCPP Milestone M5+90 Initial Availability
10/23/2007	1.0	Major	Updated and revised the technical content.
11/30/2007	1.0.1	Editorial	Revised and edited the technical content.
01/25/2008	1.0.2	Editorial	Revised and edited the technical content.
03/14/2008	1.0.3	Editorial	Revised and edited the technical content.
05/16/2008	1.0.4	Editorial	Revised and edited the technical content.
06/20/2008	2.0	Major	Updated and revised the technical content.
07/25/2008	2.1	Minor	Updated the technical content.
08/29/2008	2.1.1	Editorial	Revised and edited the technical content.
10/24/2008	2.1.2	Editorial	Revised and edited the technical content.
12/05/2008	3.0	Major	Updated and revised the technical content.
01/16/2009	3.0.1	Editorial	Revised and edited the technical content.
02/27/2009	3.0.2	Editorial	Revised and edited the technical content.
04/10/2009	4.0	Major	Updated and revised the technical content.
05/22/2009	4.1	Minor	Updated the technical content.
07/02/2009	4.1.1	Editorial	Revised and edited the technical content.
08/14/2009	4.2	Minor	Updated the technical content.
09/25/2009	4.3	Minor	Updated the technical content.
11/06/2009	4.4	Minor	Updated the technical content.
12/18/2009	5.0	Major	Updated and revised the technical content.
01/29/2010	5.1	Minor	Updated the technical content.
03/12/2010	5.2	Minor	Updated the technical content.
04/23/2010	5.2.1	Editorial	Revised and edited the technical content.
06/04/2010	5.3	Minor	Updated the technical content.
07/16/2010	5.3	No change	No changes to the meaning, language, or formatting of the technical content.
08/27/2010	5.3	No change	No changes to the meaning, language, or formatting of

Date	Revision History	Revision Class	Comments
			the technical content.
10/08/2010	5.3	No change	No changes to the meaning, language, or formatting of the technical content.
11/19/2010	6.0	Major	Significantly changed the technical content.
01/07/2011	6.0	No change	No changes to the meaning, language, or formatting of the technical content.
02/11/2011	6.0	No change	No changes to the meaning, language, or formatting of the technical content.
03/25/2011	7.0	Major	Significantly changed the technical content.
05/06/2011	7.0	No change	No changes to the meaning, language, or formatting of the technical content.
06/17/2011	7.1	Minor	Clarified the meaning of the technical content.
09/23/2011	7.1	No change	No changes to the meaning, language, or formatting of the technical content.
12/16/2011	8.0	Major	Significantly changed the technical content.
03/30/2012	8.0	No change	No changes to the meaning, language, or formatting of the technical content.
07/12/2012	8.0	No change	No changes to the meaning, language, or formatting of the technical content.
10/25/2012	8.0	No change	No changes to the meaning, language, or formatting of the technical content.
01/31/2013	8.0	No change	No changes to the meaning, language, or formatting of the technical content.
08/08/2013	9.0	Major	Significantly changed the technical content.

Contents

1 Introduction 6
1.1 Glossary 6
1.2 References 6
1.2.1 Normative References 7
1.2.2 Informative References 7
1.3 Overview 7
1.4 Relationship to Other Protocols 7
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 Common Data Types 10
2.2.1 Simple Data Types 10
2.2.1.1 BROWSER_IDENTIFY_HANDLE 10
2.2.2 Constants 10
2.2.2.1 Platform IDs 10
2.2.3 Structures 11
2.2.3.1 SERVER_INFO_100_CONTAINER 11
2.2.3.2 SERVER_ENUM_STRUCT 11

3 Protocol Details 13
3.1 Server Details 13
3.1.1 Abstract Data Model 13
3.1.1.1 OtherDomains Name Abstract Data Model 13
3.1.2 Timers 13
3.1.3 Initialization 13
3.1.4 Message Processing Events and Sequencing Rules 13
3.1.4.1 Browser 13
3.1.4.1.1 I_BrowserrQueryOtherDomains (Opnum 2) 14
3.1.5 Timer Events 15
3.1.6 Other Local Events 15
3.2 Client Details 15
3.2.1 Abstract Data Model 15
3.2.2 Timers 15
3.2.3 Initialization 16
3.2.4 Message Processing Events and Sequencing Rules 16
3.2.5 Timer Events 16
3.2.6 Other Local Events 16

4 Protocol Examples 17

5 Security 18
5.1 Security Considerations for Implementers 18
5.2 Index of Security Parameters 18

6 Appendix A: Full IDL 19

7 Appendix B: Product Behavior 21

8 Change Tracking 23

9 Index 25

1 Introduction

This document specifies the Common Internet File System (CIFS) Browser Auxiliary Protocol Specification. This protocol is used by the **master browser server** and **domain master browser server** as defined in [MS-BRWS]. The master browser server uses this protocol to query configuration information for the **domains** from the domain master browser server. The protocol operation is stateless.

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]:

browser
browser server
client
domain
domain controller (DC)
endpoint
handle
Interface Definition Language (IDL)
Internet host name
master browser server
named pipe
NetBIOS host name
opnum
primary domain controller (PDC)
server
Unicode
universally unique identifier (UUID)

The following terms are specific to this document:

domain master browser server: A **master browser server** that is responsible for combining information for an entire **domain** across all **subnets**. A **domain master browser server** is responsible for keeping multiple **subnets** in synchronization by periodically querying the **local master browser server** for information about user accounts, security, and available resources, such as printers.

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.

[C706] The Open Group, "DCE 1.1: Remote Procedure Call", C706, August 1997, http://www.opengroup.org/public/pubs/catalog/c706.htm

[MS-DTYP] Microsoft Corporation, "Windows Data Types".

[MS-RPCE] Microsoft Corporation, "Remote Procedure Call Protocol Extensions".

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

[RFC1001] Network Working Group, "Protocol Standard for a NetBIOS Service on a TCP/UDP Transport: Concepts and Methods", STD 19, RFC 1001, March 1987, http://www.ietf.org/rfc/rfc1001.txt

[RFC1002] Network Working Group, "Protocol Standard for a NetBIOS Service on a TCP/UDP Transport: Detailed Specifications", STD 19, RFC 1002, March 1987, http://www.ietf.org/rfc/rfc1002.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

[MS-BRWS] Microsoft Corporation, "Common Internet File System (CIFS) Browser Protocol".

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

[MS-WKST] Microsoft Corporation, "Workstation Service Remote Protocol".

[PIPE] Microsoft Corporation, "Named Pipes", http://msdn.microsoft.com/en-us/library/aa365590.aspx

1.3 Overview

The main objective of the CIFS Browser Auxiliary Protocol is to provide a method for the master browser server of a subnet to query specific additional information from the domain master browser server for a given domain. Selection of the master browser server and domain master browser server and the roles that these **servers** play are as specified in [MS-BRWS].

1.4 Relationship to Other Protocols

This protocol depends on RPC, as specified in [MS-RPCE], for its transport. This protocol uses RPC over **named pipes**, as specified in [MS-RPCE] section 2.1.1.2. Named pipes use the Server Message Block (SMB) Protocol, as specified in [MS-SMB].

An implementation of [MS-BRWS] may use this protocol to retrieve information from the domain master browser.

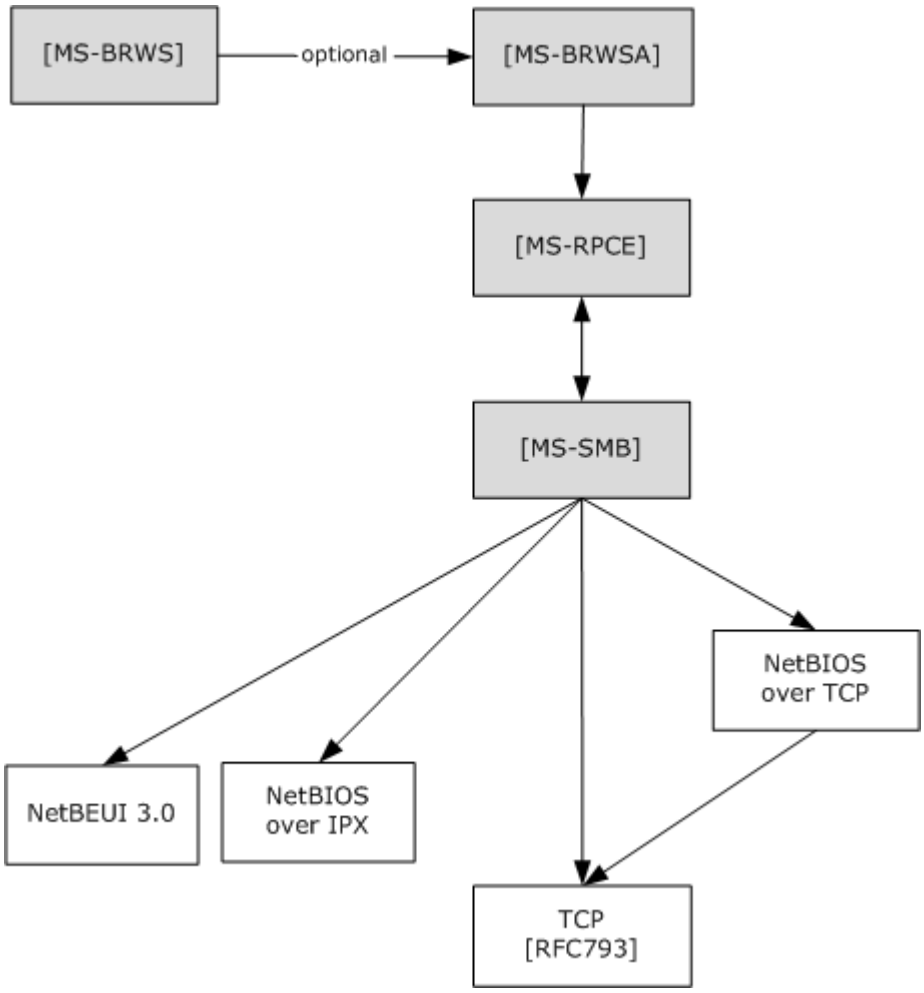

Figure 1: Relationship to other protocols

- [MS-BRWS] calls (optional) [MS-BRWSA] to request OtherDomains information from a **domain controller (DC)**.
- [MS-BRWSA] calls [MS-RPCE] as RPC/named pipes as transport.
- [MS-RPCE] calls [MS-SMB] named pipes as a transport that uses SMB.

1.5 Prerequisites/Preconditions

The master browser server has previously identified the **endpoint** address of the domain master browser server.

1.6 Applicability Statement

This protocol is used to retrieve the list of domains that the domain master browser server has been configured to support.

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

Parameter	Value	Reference
RPC Interface UUID	6BFFD098-A112-3610-9833-012892020162	See section 2.1
Pipe Name	"\pipe\browser"	See section 2.1

2 Messages

2.1 Transport

The RPC methods that the CIFS Browser Auxiliary Protocol uses are available on one endpoint:

- "\pipe\browser" named pipe (RPC protseqs ncacn_np), as specified in [MS-RPCE] section 2.1.1.2.

The CIFS Browser Auxiliary Protocol endpoint is available only over named pipes. For more information about named pipes, see [PIPE].

This protocol MUST use the **universally unique identifier (UUID)** as specified in section 1.9. The RPC version number is 0.0.

This protocol allows any user to establish a connection to the RPC server. The protocol uses the underlying RPC protocol to retrieve the identity of the caller that made the method call, as described in section 3.3.3.4.3 of [MS-RPCE]. The server SHOULD use this identity to perform method specific access checks as described in section 3.1.4.

2.2 Common Data Types

In addition to RPC base types and definitions specified in [C706] and [MS-RPCE], additional data types are defined below.

The following are the types that are defined in this specification.

2.2.1 Simple Data Types

2.2.1.1 BROWSER_IDENTIFY_HANDLE

The **BROWSER_IDENTIFY_HANDLE** structure is a null-terminated **Unicode** string that identifies the remote computer on which to execute the method.

This type is declared as follows:

```
typedef [handle] LPWSTR BROWSER_IDENTIFY_HANDLE;
```

The **client** MUST set the impersonation level for the RPC connection that refers to this **handle** to "IDENTIFICATION". "IDENTIFICATION" implies an impersonation level of SECURITY_IDENTIFICATION. For more information on impersonation levels, see the **ImpersonationLevel** field in [MS-SMB] section 2.2.4.9.1.

2.2.2 Constants

2.2.2.1 Platform IDs

The following values specify the information level to use for platform-specific information on the server.

Name	Value (decimal)
PLATFORM_ID_DOS	300
PLATFORM_ID_OS2	400
PLATFORM_ID_NT	500
PLATFORM_ID_OSF	600
PLATFORM_ID_VMS	700

2.2.3 Structures

2.2.3.1 SERVER_INFO_100_CONTAINER

The **SERVER_INFO_100_CONTAINER** structure contains a count of the entries returned by the method and a pointer to a buffer.

```
typedef struct _SERVER_INFO_100_CONTAINER {
  DWORD EntriesRead;
  [size_is(EntriesRead)] LPSERVER_INFO_100 Buffer;
} SERVER_INFO_100_CONTAINER,
 *PSERVER_INFO_100_CONTAINER,
 *LPSERVER_INFO_100_CONTAINER;
```

EntriesRead: The number of entries returned by the method call. This value MUST be zero if no domains are configured in the **primary domain controller** or domain controller. The client SHOULD set the EntriesRead field to 0, and the Buffer field to NULL, and the server MUST ignore these fields.

Buffer: A pointer to an array of SERVER_INFO_100 data structures (as specified in [MS-DTYP] section 2.3.11). If EntriesRead is zero, this field is undefined and MUST NOT be considered a valid pointer.

2.2.3.2 SERVER_ENUM_STRUCT

The **SERVER_ENUM_STRUCT** structure defines the layout for a structure with a value to indicate the information level submitted to the method and a pointer to a data structure that contains an array of data structures returned by the method. This structure is used by I_BrowserrQueryOtherDomains.

```
typedef struct _SERVER_ENUM_STRUCT {
  DWORD Level;
  [switch_is(Level)] union {
    [case(100)]
      LPSERVER_INFO_100_CONTAINER Level100;
    [default]      ;
  } ServerInfo;
} SERVER_ENUM_STRUCT,
 *PSERVER_ENUM_STRUCT,
 *LPSERVER_ENUM_STRUCT;
```

Level: The information level of the data. This member MUST be 100.

ServerInfo: A structure that contains an array of data structures. The **Level** member determines the data type of the members of this array.

Level100: A pointer to a **SERVER_INFO_100_CONTAINER** structure that contains the number of entries returned by the method and a pointer to an array of SERVER_INFO_100 structures (as specified in [MS-DTYP] section 2.3.11).

3 Protocol Details

3.1 Server Details

3.1.1 Abstract Data Model

3.1.1.1 OtherDomains Name Abstract Data Model

OtherDomains: Specifies a list of NetBIOS names of domains, as specified in [RFC1001] and [RFC1002], browsed by the computer. Each name MUST be at most 15 characters in length, and MUST NOT contain trailing spaces or a NetBIOS suffix as defined in [MS-BRWS] section 2.1.1. The names in the **OtherDomains** list MUST be separated by spaces.

This element is shared with the Workstation Service Remote Protocol Specification [MS-WKST], queried through the WkstaQueryOtherDomains event (section 3.2.6.1).

The **OtherDomains** element is also shared with the Common Internet File System (CIFS) Browser Protocol [MS-BRWS] to update the **OtherDomains** information from a domain controller.

3.1.2 Timers

None.

3.1.3 Initialization

Section 2.1 specifies the parameters necessary to initialize the RPC protocol.

3.1.4 Message Processing Events and Sequencing Rules

The *ServerName* parameter MUST be ignored by the server when processing any message.<1>

3.1.4.1 Browser

The **browser** interface lists the methods associated with the **browser** service, which creates and maintains a view of resources available on a network. The server does not maintain client state information. The protocol operation is stateless. The version for this interface is 0.0.

The UUID for this interface is: "6BFFD098-A112-3610-9833-012892020162".

Methods in RPC Opnum Order

Method	Description
Opnum0NotUsedOnWire	Reserved for local use. Opnum: 0
Opnum1NotUsedOnWire	Reserved for local use. Opnum: 1
I_BrowserrQueryOtherDomains	Returns a list of other domains configured for this computer. Opnum: 2
Opnum3NotUsedOnWire	Reserved for local use.

Method	Description
	Opnum: 3
Opnum4NotUsedOnWire	Reserved for local use. Opnum: 4
Opnum5NotUsedOnWire	Reserved for local use. Opnum: 5
Opnum6NotUsedOnWire	Reserved for local use. Opnum: 6
Opnum7NotUsedOnWire	Reserved for local use. Opnum: 7
Opnum8NotUsedOnWire	Reserved for local use. Opnum: 8
Opnum9NotUsedOnWire	Reserved for local use. Opnum: 9
Opnum10NotUsedOnWire	Reserved for local use. Opnum: 10
Opnum11NotUsedOnWire	Reserved for local use. Opnum: 11

In the preceding table, the phrase "Reserved for local use" means that the client MUST NOT send the **opnum** and that the server behavior is undefined<2> because it does not affect interoperability.

3.1.4.1.1 I_BrowserrQueryOtherDomains (Opnum 2)

The **I_BrowserrQueryOtherDomains** method is received by the server in an RPC_REQUEST packet. The client SHOULD NOT send this RPC request to a server that is not a primary domain controller (PDC) acting as the **domain master browser server**.

If this server is not a primary domain controller it MAY fail the request.<3>

If the server is a primary domain controller, the server MUST update **OtherDomains** as specified in [MS-WKST] section 3.2.6.1, WkstaQueryOtherDomains Event. The server MUST construct a SERVER_ENUM structure as specified in 2.2.3.2, containing a SERVER_INFO_100 structure as specified in [MS-DTYP] section 2.3.11 for each name in **OtherDomains**, and return this to the caller.

```
NET_API_STATUS I_BrowserrQueryOtherDomains(
  [in, string, unique] BROWSER_IDENTIFY_HANDLE ServerName,
  [in, out] LPSERVER_ENUM_STRUCT InfoStruct,
  [out] LPDWORD TotalEntries
);
```

ServerName: An optional **BROWSER_IDENTIFY_HANDLE** structure that specifies the name of the server to execute the method. This value is ignored upon receipt.

InfoStruct: A pointer to a **SERVER_ENUM_STRUCT** structure that contains the **Level** member and a pointer to a SERVER_INFO_x structure, where <x> MUST be 100. The **Level** member MUST be set to 100. If the **Level** member is set to any other value, the method MUST return ERROR_INVALID_LEVEL.<4>

TotalEntries: The number of entries returned by the method call. This parameter MUST match the **EntriesRead** member of the **SERVER_INFO_100_CONTAINER** structure.

Return Values: The method returns NERR_Success on success; otherwise, it returns a nonzero error code, as specified in either Win32 Error Codes. The most common error codes are listed in the following table.<5>

Return value/code	Description
0x00000000 NERR_Success	The operation completed successfully.
0x00000005 ERROR_ACCESS_DENIED	Access is denied.
0x00000008 ERROR_NOT_ENOUGH_MEMORY	This value MUST be returned when the server could not allocate enough memory to complete this operation.
0x00000057 ERROR_INVALID_PARAMETER	This value MUST be returned when a parameter is incorrect. For example, this value is returned when the *InfoStruct* parameter is NULL or the **Level100** member in the structure pointed to by the *InfoStruct* parameter is NULL.
0x0000007C ERROR_INVALID_LEVEL	This value MUST be returned when the **Level** member is not 100.
0x000000EA ERROR_MORE_DATA	The error ERROR_MORE_DATA indicates that not all available entries were returned. Some more entries exist which were not returned in the response.

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

3.2 Client Details

3.2.1 Abstract Data Model

None.

3.2.2 Timers

None.

3.2.3 Initialization

The client MUST create an RPC connection to the remote computer by using details as specified in section 2.1.

3.2.4 Message Processing Events and Sequencing Rules

None.

3.2.5 Timer Events

None.

3.2.6 Other Local Events

None.

4 Protocol Examples

The method provided by this protocol is a simple request-response. The server receives the request, executes the method, and returns a completion. The client simply returns the completion status to the calling application. This is a stateless protocol; each method call is independent of any previous method calls.

For example, the client calls the **I_BrowserrQueryOtherDomains** method; on receiving this method, the server executes the call locally and returns the appropriate information and NERR_Success.

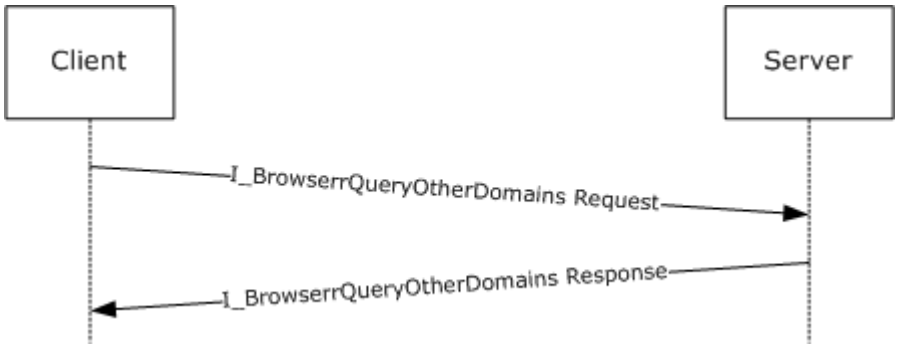

Figure 2: Simple request-response model

5 Security

5.1 Security Considerations for Implementers

As described in section 2.1, this protocol allows any user to connect to the server. Therefore, any security bug in the server implementation could be exploitable. The server implementation should enforce security on each method.

5.2 Index of Security Parameters

None.

6 Appendix A: Full IDL

For ease of implementation, the full **IDL** is provided below, where "ms-dtyp.idl" refers to the IDL found in [MS-DTYP] Appendix A.

The syntax uses the IDL syntax extensions defined in [MS-RPCE] sections 2.2.4 and 2.2.4. For example, as noted in [MS-RPCE] section 2.2.4.9, a pointer_default declaration is not required and pointer_default(unique) is assumed.

```
[
    uuid(6BFFD098-A112-3610-9833-012892020162),
    version(0.0),
    ms_union,
    pointer_default(unique)
]
interface browser
{
    import "ms-dtyp.idl";

   typedef WCHAR* LPWSTR;

    typedef [handle] LPWSTR BROWSER_IDENTIFY_HANDLE;

   typedef struct _SERVER_INFO_100_CONTAINER {
        DWORD   EntriesRead;
        [size_is(EntriesRead)] LPSERVER_INFO_100 Buffer;
     } SERVER_INFO_100_CONTAINER,
      *PSERVER_INFO_100_CONTAINER,
      *LPSERVER_INFO_100_CONTAINER;

    typedef struct _SERVER_ENUM_STRUCT {
        DWORD   Level;
        [switch_is(Level)] union _SERVER_ENUM_UNION {
            [case(100)]
                LPSERVER_INFO_100_CONTAINER Level100;
            [default]
                ;
        } ServerInfo;
    } SERVER_ENUM_STRUCT,
     *PSERVER_ENUM_STRUCT,
     *LPSERVER_ENUM_STRUCT;


    NET_API_STATUS Opnum0NotUsedOnWire(void);

    NET_API_STATUS Opnum1NotUsedOnWire(void);

    NET_API_STATUS
    I_BrowserrQueryOtherDomains(
        [in,string,unique] BROWSER_IDENTIFY_HANDLE ServerName,
        [in,out]           LPSERVER_ENUM_STRUCT    InfoStruct,
        [out]              LPDWORD                 TotalEntries
    );

    NET_API_STATUS Opnum3NotUsedOnWire(void);

    NET_API_STATUS Opnum4NotUsedOnWire(void);
```

```
        NET_API_STATUS Opnum5NotUsedOnWire(void);

        NET_API_STATUS Opnum6NotUsedOnWire(void);

        NET_API_STATUS Opnum7NotUsedOnWire(void);

        NET_API_STATUS Opnum8NotUsedOnWire(void);

        NET_API_STATUS Opnum9NotUsedOnWire(void);

        NET_API_STATUS Opnum10NotUsedOnWire(void);

        NET_API_STATUS Opnum11NotUsedOnWire(void);
    }
```

7 Appendix B: 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 2000 operating system
- Windows XP operating system
- Windows Server 2003 operating system
- 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 3.1.4: Windows implementation uses the RPC protocol to retrieve the identity of the caller as described in section 3.3.3.4.3 of [MS-RPCE]. The server uses the underlying Windows security subsystem to determine the permissions for the caller. If the caller does not have the required permissions to execute a specific method, the method call fails with ERROR_ACCESS_DENIED (0x00000005).

<2> Section 3.1.4.1: Opnums reserved for local use apply to Windows as follows:

opnum	Description
0,5,6,10,11	Only used locally by Windows, never remotely.
1,3,4,9	Just returns ERROR_NOT_SUPPORTED. It is never used.
7,8	Just returns NERR_Success. It is never used.

<3> Section 3.1.4.1.1: Windows servers accept the **I_BrowserrQueryOtherDomains (section 3.1.4.1.1)** request and return the names of domains configured for browsing even if they are not the primary domain controller.

<4> Section 3.1.4.1.1: PLATFORM_ID_OS2 should be used for Windows NT Server, Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, and Windows Server 2012 R2.

<5> Section 3.1.4.1.1: If the caller is not an authenticated user, then the method call fails with ERROR_ACCESS_DENIED (0x00000005).

8 Change Tracking

This section identifies changes that were made to the [MS-BRWSA] 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
7 Appendix B: Product Behavior	Modified this section to include references to Windows 8.1 operating system and Windows Server 2012 R2 operating system.	Y	Content updated.

9 Index

A

Abstract data model
  client 15
  server - OtherDomains 13
Applicability 8

B

Browser method 13
BROWSER_IDENTIFY_HANDLE simple type 10

C

Capability negotiation 9
Change tracking 23
Client
  abstract data model 15
  initialization 16
  local events 16
  message processing 16
  sequencing rules 16
  timer events 16
  timers 15
Common data types - overview 10
Constants - platform IDs 10

D

Data model - abstract
  client 15
  server - OtherDomains 13
Data types
  BROWSER_IDENTIFY_HANDLE simple type 10
  common 10
  simple - BROWSER_IDENTIFY_HANDLE 10

E

Examples - overview 17

F

Fields - vendor-extensible 9
Full IDL 19

G

Glossary 6

I

I_BrowserrQueryOtherDomains method 14
IDL 19
Implementer - security considerations 18
Index of security parameters 18
Informative references 7
Initialization
  client 16
  server 13
Introduction 6

L

Local events
  client 16
  server 15
LPSERVER_ENUM_STRUCT 11
LPSERVER_INFO_100_CONTAINER 11

M

Message processing
  client 16
  server 13
Messages
  platform IDs constant 10
  transport 10
Methods - browser 13

N

Normative references 7

O

Overview (synopsis) 7

P

Parameters - security index 18
Platform IDs constant 10
Preconditions 8
Prerequisites 8
Product behavior 21
PSERVER_ENUM_STRUCT 11
PSERVER_INFO_100_CONTAINER 11

R

References
  informative 7
  normative 7
Relationship to other protocols 7

S

Security
  implementer considerations 18
  parameter index 18
Sequencing rules
  client 16
  server 13
Server
  abstract data model - OtherDomains 13
  browser method 13

initialization 13
local events 15
message processing 13
sequencing rules 13
timer events 15
timers 13
SERVER_ENUM_STRUCT structure 11
SERVER_INFO_100_CONTAINER structure 11
Simple data types - BROWSER_IDENTIFY_HANDLE 10
Standards assignments 9

T

Timer events
client 16
server 15
Timers
client 15
server 13
Tracking changes 23
Transport 10

V

Vendor-extensible fields 9
Versioning 9