
 Filter by title
(Opnum 12)
3.1.4.6 NetrSessionDel (Opnum 13)
3.1.4.7 NetrShareAdd (Opnum 14)
3.1.4.8 NetrShareEnum (Opnum 15)
3.1.4.9 NetrShareEnumSticky (Opnum 36)
3.1.4.10 NetrShareGetInfo (Opnum 16)
3.1.4.11 NetrShareSetInfo (Opnum 17)
3.1.4.12 NetrShareDel (Opnum 18)
3.1.4.13 NetrShareDelSticky (Opnum 19)
3.1.4.14 NetrShareDelStart (Opnum 37)
3.1.4.15 NetrShareDelCommit (Opnum 38)
3.1.4.16 NetrShareCheck (Opnum 20)
3.1.4.17 NetrServerGetInfo (Opnum 21)
3.1.4.18 NetrServerSetInfo (Opnum 22)
3.1.4.19 NetrServerDiskEnum (Opnum 23)
3.1.4.20 NetrServerStatisticsGet (Opnum 24)
3.1.4.21 NetrRemoteTOD (Opnum 28)
3.1.4.22 NetrServerTransportAdd
 Download PDF

Learn /



3.1.4.5 NetrSessionEnum (Opnum 12)

Article • 04/07/2021

 Feedback

The NetrSessionEnum method MUST return information about sessions that are established on a [server](#) or return an error code.

```
NET_API_STATUS NetrSessionEnum(  
    [in, string, unique] SRVSVC_HANDLE ServerName,  
    [in, string, unique] WCHAR* ClientName,  
    [in, string, unique] WCHAR* UserName,  
    [in, out] PSESSION_ENUM_STRUCT InfoStruct,  
    [in] DWORD PreferredMaximumLength,  
    [out] DWORD* TotalEntries,  
    [in, out, unique] DWORD* ResumeHandle  
);
```

ServerName: An [SRVSVC_HANDLE \(section 2.2.1.1\)](#) pointer that identifies the server. The [client](#) MUST map this structure to an RPC binding handle (see [\[C706\] ↗](#) sections 4.3.5 and 5.1.5.2). The server MUST ignore this parameter.

ClientName: A pointer to a null-terminated UTF-16 string that specifies the name of the computer session for which information is to be returned. This string MUST be one of the following: a NULL (zero-length) string; or a string that MUST begin with \\.

UserName: A pointer to a null-terminated UTF-16 string that specifies the user name for which information is to be returned.

InfoStruct: A pointer to a structure, in the format of a [SESSION_ENUM_STRUCT](#). The SESSION_ENUM_STRUCT structure has a **Level** member that specifies the type of structure to return. The **Level** member MUST be one of the values specified in section 2.2.4.21.


PreferredMaximumLength: Specifies the preferred maximum length, in bytes, of the returned data. If the value that is specified is [MAX_PREFERRED_LENGTH](#), the method MUST attempt to return all entries.

TotalEntries: The total number of entries that could have been enumerated if the buffer had been big enough to hold all the entries.

ResumeHandle: A pointer to a value that contains a handle that is used to continue an existing session search in **SessionList**, as specified in section [3.1.1.1](#). The handle MUST be zero on the first call and remain unchanged for subsequent calls. If the *ResumeHandle* parameter is NULL, no resume handle MUST be stored. If this parameter is not NULL and the method returns ERROR_MORE_DATA, this parameter receives an implementation-specific nonzero value that can be passed in subsequent calls to this method to continue with the enumeration.

If this parameter is NULL or points to 0x00000000, the enumeration starts from the beginning of the **SessionList**.

Return Values: The method returns 0x00000000 (NERR_Success) to indicate success; otherwise, it returns a nonzero error code. The method can take any specific error code value, as specified in [\[MS-ERREF\]](#) section [2.2](#). The most common error codes are listed in the following table.

 Expand table

Return value/code	Description
0x00000000 NERR_Success	The client request succeeded.
0x00000005 ERROR_ACCESS_DENIED	Access is denied.
0x0000007C ERROR_INVALID_LEVEL	The system call level is not correct.
0x00000057 ERROR_INVALID_PARAMETER	The parameter is incorrect.
0x000000EA ERROR_MORE_DATA	The client request succeeded. More entries are available. Not all entries could be returned in the buffer size that is specified by <i>PreferedMaximumLength</i> .
0x00000008 ERROR_NOT_ENOUGH_MEMORY	Not enough storage is available to process this command.
0x00000908 NERR_ClientNameNotFound	A session does not exist with the computer name.
0x0000092F NERR_InvalidComputer	The computer name is not valid.
0x000008AD NERR_UserNotFound	The user name could not be found.

In response to the NetrSessionEnum message, the server MUST enumerate the **Session** entries in **SessionList** based on the value of the *ResumeHandle* parameter. For each entry, the server MUST query session properties by invoking the underlying server events as specified in [\[MS-CIFS\]](#) section 3.3.4.14 and [\[MS-SMB2\]](#) section 3.3.4.18, providing *Session.GlobalSessionId* as the input parameter. When the server receives a STATUS SUCCESS for a *Session.GlobalSessionId* from either a CIFS or SMB2 server, the server MUST consider the received SESSION_INFO_502 structure as valid, and it MUST continue to query all other sessions that are established on the server. The server MUST then return information about some or all valid sessions that are established on the server, depending on the qualifier parameters that are specified.

The *ClientName* parameter specifies a qualifier for the returned information. If a *ClientName* is specified (that is, it is not a NULL (zero-length) string), the sesi502_cname field returned in the SESSION_INFO_502 structure MUST match the *ClientName* for the session to be returned.

If a *ClientName* is specified, it MUST start with "\\"; otherwise, the server MUST fail the call with a NERR_InvalidComputer error code. If a *ClientName* is specified and it contains more than 1,024 characters, including the terminating null character, the server MUST fail the call with an ERROR_INVALID_PARAMETER error code.

The *UserName* parameter specifies a qualifier for the returned information. If a *UserName* is specified (that is, not a NULL (zero-length) string), the sesi502_username field returned in the SESSION_INFO_502 structure MUST match the *UserName* parameter for the session to be returned. If a *UserName* parameter is specified and the length of the *UserName* string, including the terminating null character, is greater than 1,024 characters, the server MUST fail the call with an ERROR_INVALID_PARAMETER error code.

The server MUST return only those sessions that match all specified qualifiers. If no entries that match the qualifiers (ClientName/UserName) are found when a qualifier is specified, the server MUST fail the call with either an NERR_UserNotFound or NERR_ClientNameNotFound error code.

The *ClientName* and *UserName* parameters have no role in determining the value of *ResumeHandle*. The server uses the *ResumeHandle* parameter to start the enumeration (as described in the processing rules that follow for the *ResumeHandle* parameter), and then applies these qualifier parameters, if specified, to restrict the returned results to only those items that pass the qualifier test (as described previously in this topic for *ResumeHandle*).

The *InfoStruct* parameter has a **Level** member whose valid values are 0, 1, 2, 10, and 502. If the **Level** member is not equal to one of the valid values, the server MUST fail the call with an ERROR_INVALID_LEVEL error code.

The server MUST fill the return structures as follows.

If the **Level** member is 0, the server MUST return the information about sessions by filling the [SESSION_INFO_0_CONTAINER](#) structure in the **SessionInfo** field of the *InfoStruct* parameter as follows. The SESSION_INFO_0_CONTAINER structure contains an array of [SESSION_INFO_0](#) structures.

- **sesi0_cname** MUST be set to **session.sesi502_cname**.

If the **Level** member is 1, the server MUST return the information about sessions by filling the [SESSION_INFO_1_CONTAINER](#) structure in the **SessionInfo** field of the *InfoStruct* parameter as in the following. The SESSION_INFO_1_CONTAINER structure contains an array of [SESSION_INFO_1](#) structures.

- **sesi1_cname** MUST be set to **session.sesi502_cname**.
- **sesi1_username** MUST be set to **session.sesi502_username**.
- **sesi1_num_opens** MUST be set to **session.sesi502_num_opens**.

If the **Level** member is 2, the server MUST return the information about **sessions** by filling the [SESSION_INFO_2_CONTAINER](#) structure in the **SessionInfo** field of the *InfoStruct* parameter as in the following. The SESSION_INFO_2_CONTAINER structure contains an array of [SESSION_INFO_2](#) structures.

- **sesi2_cname** MUST be set to **session.sesi502_cname**.
- **sesi2_username** MUST be set to **session.sesi502_username**.
- **sesi2_num_opens** MUST be set to **session.sesi502_num_opens**.
- **sesi2_idle_time** MUST be set to **session.sesi502_idletime**.
- **sesi2_time** MUST be set to **session.sesi502_time**.
- **sesi2_user_flags** MUST be set to **session.sesi502_user_flags**.
- **sesi2_cltype_name** MUST be set to **session.sesi502_cltype_name**.

If the **Level** member is 10, the server MUST return the information about **sessions** by filling the [SESSION_INFO_10_CONTAINER](#) structure in the **SessionInfo** field of the *InfoStruct* parameter as in the following. The SESSION_INFO_10_CONTAINER structure contains an array of [SESSION_INFO_10](#) structures.

- **sesi10_cname** MUST be set to **session.sesi502_cname**.
- **sesi10_username** MUST be set to **session.sesi502_username**.
- **sesi10_idle_time** MUST be set to **session.sesi502_idletime**.
- **sesi10_time** MUST be set to **session.sesi502_time**.

If the **Level** member is 502, the server MUST return the sessions in the [SESSION_INFO_502](#) structure by filling the [SESSION_INFO_502_CONTAINER](#) structure in the **SessionInfo** field of the *InfoStruct* parameter. The SESSION_INFO_502_CONTAINER structure contains an array of SESSION_INFO_502 structures.

The *PreferedMaximumLength* parameter specifies the maximum number of bytes that the server can return for the **SessionInfo** buffer. If *PreferedMaximumLength* is insufficient to hold all the entries, the server MUST return the maximum number of entries that will fit in the **SessionInfo** buffer and return ERROR_MORE_DATA. If this parameter is equal to MAX_PREFERRED_LENGTH, the server MUST return all the requested data.

If the server returns NERR_Success or ERROR_MORE_DATA, it MUST set the *TotalEntries* parameter to equal the total number of entries that exceed the qualifier filter (*ClientName* or *UserName* as previously described) and that could have been enumerated from the current resume position.

If the *PreferedMaximumLength* is insufficient to hold all the entries and if the client has specified a *ResumeHandle*, the server MUST set *ResumeHandle* to some implementation-specific value that allows the server to continue with this enumeration on a subsequent call to this method with the same value for *ResumeHandle*.

The following rules specify processing of the *ResumeHandle* parameter:

- If the *ResumeHandle* parameter is either NULL or points to 0x00000000, the enumeration MUST start from the beginning of the **SessionList**.
- If the *ResumeHandle* parameter points to a nonzero value, the server must validate the *ResumeHandle*.
 - If the value of *ResumeHandle* is less than the size of the **SessionList**, the server MUST continue enumeration based on the value of *ResumeHandle*. The value of *ResumeHandle* specifies the index into the **SessionList** after which enumeration is to begin.
 - If the value of *ResumeHandle* is greater than or equal to the size of the **SessionList**, the server MUST return NERR_Success and zero entries.
- If the client specified a *ResumeHandle* and the server returns ERROR_MORE_DATA (0x000000EA), the server MUST set *ResumeHandle* to the index value of the last enumerated session in the **SessionList**.

Because the *ResumeHandle* specifies the index into the list and the list of active sessions can be modified between multiple requests, the results of a query spanning multiple requests using the *ResumeHandle* can be unreliable, offering either duplicate or inactive sessions.

The server SHOULD<52> enforce the security measures to verify that the caller has the required permissions to execute this routine. If the caller does not have the required credentials, the server SHOULD<53> fail the call.

Additional resources

Events

Nov 20, 12 AM - Nov 22, 12 AM

Join online sessions at Microsoft Ignite created to expand your skills and help you tackle today's complex issues.

[Register now](#)