SR 118100419160002 joining readonly domain controller not documented in MS-WKST

This is regarding provisional draft changes (in red highlights) related to [MS-WKST] 3.2.4.13 NETSETUP_JOIN_READONLY option in NetrJoinDomain2

3.2.4.13 NetrJoinDomain2 (Opnum 22)

The **NetrJoinDomain2** method uses encrypted credentials to join a computer to a **domain** or a workgroup.<56>

For high-level, informative discussions about **domain controller** location and domain join and unjoin, see [MS-ADOD] section 2.7.7 and [MS-ADOD] section 3.1. Also, see the example in section 4.3 for more information.

```
unsigned long NetrJoinDomain2(
  [in] handle_t RpcBindingHandle,
  [in, string, unique] wchar_t* ServerName,
  [in, string] wchar_t* DomainNameParam,
  [in, string, unique] wchar_t* MachineAccountOU,
  [in, string, unique] wchar_t* AccountName,
  [in, unique] PJOINPR_ENCRYPTED_USER_PASSWORD Password,
  [in] unsigned long Options
);
```

RpcBindingHandle: An **RPC** binding **handle** [C706].

ServerName: This parameter has no effect on message processing in any environment. The client MUST set this parameter to a value that resolves to the IP protocol layer destination address of the RPC packets it transmits ([MS-RPCE] section 2.1.1.2). The **server** MUST ignore this parameter.

DomainNameParam: A pointer to a string that specifies the **domain name** or workgroup name to join, and optionally the domain controller machine name within the domain. This parameter MUST NOT be NULL.

If the string specifies the name of the preferred domain controller to perform the join operation, then the string MUST be of the form *DomainNameToJoin**MachineName*, where *DomainNameToJoin* is the domain to join, "\" is a delimiter, and *MachineName* is the name of the domain controller to perform the join operation. In all cases, the *DomainNameToJoin* portion of this parameter MUST be either the **NetBIOS name** of the domain or the **fully qualified domain name (FQDN)** of the domain. If the *MachineName* is passed, it MUST be either the NetBIOS name of the domain controller or the **Internet host name** of the domain controller. The format of *DomainNameToJoin* places no constraint on the format of *MachineName* and vice versa; thus, each of the following permutations are accepted:

- NetBIOS name\NetBIOS name
- NetBIOS name\Internet host name
- FQDN\NetBIOS name

MachineAccountOU: A pointer to a string that MUST contain [RFC1777] the format name of the **organizational unit (OU)** directory object under which the **machine account** directory object is created. This parameter is optional. If specified, this string MUST contain the full path; for example, OU=testOU,DC=domain,DC=Domain,DC=com.

AccountName: A pointer to a string that specifies an account name in the domain *DomainNameParam* to use when connecting to a domain controller. This parameter is optional. If this parameter is NULL, the caller's account name MUST be used. If this parameter is specified, the format MUST be one of the following:

- <NetBIOSDomainName>\<UserName>
- <FullyQualifiedDNSDomainName>\<UserName>
- <UserName>@<FullyQualifiedDNSDomainName>

Password: A pointer to a **JOINPR_ENCRYPTED_USER_PASSWORD** (section 2.2.5.18) structure that specifies the encrypted password to use with the *AccountName* parameter. Sections 3.2.4.13.1 and 3.2.4.13.3 specify the processing of this parameter.

Options: A 32-bit bitfield that specifies modifications to default server behavior in message processing.<57>

Value/code	Meaning
NETSETUP_JOIN_DOMAIN 0x00000001	Joins the computer to a domain. The default action is to join the computer to a workgroup.
NETSETUP_ACCT_CREATE 0x00000002	Creates the account on the domain. The name is the persisted abstract state **ComputerNameNetBIOS** unless this behavior is altered by another option such as NETSETUP_JOIN_WITH_NEW_NAME.
NETSETUP_ACCT_DELETE 0x00000004	Disables the old account when the join operation occurs on a computer that is already joined to a domain. **Important** This flag is neither supported nor tested for use with **NetrJoinDomain2**; its use is therefore not specified in any message processing.
NETSETUP_DOMAIN_JOIN_IF_JOINED 0x00000020	Allows a join to a new domain even if the computer is already joined to a domain.
NETSETUP_JOIN_UNSECURE 0x00000040	Performs an unsecured join. MUST be used only in conjunction with the NETSETUP_MACHINE_PWD_PASSED flag.
NETSETUP_MACHINE_PWD_PASSED 0x00000080	Indicates that the *Password* parameter SHOULD<58> specify the password for the machine joining the domain. This flag is valid only for unsecured joins, which MUST be indicated by setting the NETSETUP_JOIN_UNSECURE flag, or read-only joins, which MUST be indicated by setting the NETSETUP_JOIN_READONLY flag. If this flag is set, the value of *Password* determines the value stored for the computer password during the join process.
NETSETUP_DEFER_SPN_SET 0x00000100	Indicates that the **service principal name (SPN)** and the **DnsHostName** properties on the computer SHOULD NOT<59> be updated at this time, but instead SHOULD<60> be updated during a subsequent call to **NetrRenameMachineInDomain2** (section 3.2.4.15).
NETSETUP_JOIN_DC_ACCOUNT 0x00000200	Indicates that the join SHOULD<61> be allowed if an existing account exists and it is a domain controller account.<62>
NETSETUP_JOIN_WITH_NEW_NAME 0x00000400	Indicates that the join SHOULD<63>occur using the new **computer name**.
NETSETUP_JOIN_READONLY	Specifies that the join SHOULD <121> be performed in a read-only manner against an existing account object. This option is intended

Value/code	Meaning
0x00000800	to enable the server to join a domain using a read-only domain controller.
NETSETUP_INSTALL_INVOCATION 0x00040000	Indicates that the protocol method was invoked during installation.

Return Values: When the message processing result meets the description in column two of the following table, this method MUST return one of the following values ([MS-ERREF] section 2.2).

Value/code	Meaning
NERR_Success 0x00000000	The operation completed successfully.
ERROR_FILE_NOT_FOUND 0x00000002	The object was not found.
ERROR_ACCESS_DENIED 0x00000005	Access is denied.
ERROR_NOT_SUPPORTED 0x00000032	The request is not supported.
ERROR_INVALID_PASSWORD 0x00000056	The specified network password is not correct.
ERROR_INVALID_PARAMETER 0x00000057	The parameter is incorrect.
ERROR_PASSWORD_RESTRICTION 0x0000052D	Unable to update the password. The value provided for the new password does not meet the length, complexity, or history requirements of the domain.
ERROR_LOGON_FAILURE 0x0000052E	Logon failure: unknown user name or bad password.
ERROR_NONE_MAPPED 0x00000534	The account was not found.
ERROR_INVALID_DOMAIN_ROLE 0x0000054A	The name of a domain controller was provided in the *DomainNameParam* parameter, and validation of that domain controller failed. Validation is specified in the message-processing steps for the section "Domain Join" later.
ERROR_NO_SUCH_DOMAIN 0x0000054B	The specified domain either does not exist or could not be contacted.
RPC_S_PROTSEQ_NOT_SUPPORTED 0x000006A7	The **RPC protocol sequence** is not supported.
RPC_S_CALL_IN_PROGRESS 0x000006FF	A remote procedure call is already in progress.<64>
NERR_UserExists 0x000008B0	The user account already exists.
NERR_SetupAlreadyJoined 0x00000A83	This computer is already joined to a domain.

Value/code	Meaning
NERR_SetupDomainController 0x00000A85	This computer is a domain controller and cannot be unjoined from a domain.
NERR_InvalidWorkgroupName 0x00000A87	The specified workgroup name is invalid.

Any other return value MUST conform to the error code requirements specified in **Protocol Details** (section 3).

Message processing for the **NetrJoinDomain2** method specifies the behavior of joining either a domain or a workgroup. The behavior of this method is covered in the following subsections:

- Section 3.2.4.13.1 specifies the message processing that is common to both domain and workgroup joins.
- Section 3.2.4.13.2 specifies the state transition associated with a domain join.
- Section 3.2.4.13.3 specifies the message processing that is involved in a domain join.
- Section 3.2.4.13.4 specifies the message processing that is involved in a workgroup join.

Several password data elements are involved in message processing for the **NetrJoinDomain2** method, and they are distinguished as follows:

Password: A parameter to this method, either the password corresponding to the *AccountName* that is used to **authenticate** at the domain controller or the password used for the computer account. The bits in the *Options* parameter determine how *Password* is used. This element is distinct from the **client** data model element **Password** that is defined in section 3.2.1.6.

PasswordString: The **Unicode UTF-8** string that corresponds to the **plaintext** form of the password in *Password*. This variable is relevant to sections 3.2.4.13.1 and 3.2.4.13.3.

ComputerPasswordString: The **ASCII** string that contains the plaintext form of the password for the computer account. This variable is relevant to section 3.2.4.13.3.

3.2.4.13.3 Domain Join Specific Message Processing

The following definitions are used in the specification of message processing that follows.

- *DomainNameString*: A **Unicode UTF-8** string with the same properties specified for the parameter *DomainNameParam*.
- *DomainControllerString*: A UTF-8 string that contains the name of a **domain controller** in the **domain** that the **server** is joining.
- *DomainObject*: An object in the domain database ([MS-ADTS] section 6.4).
- *MachineAccountOUString*: A UTF-8 string that contains the **organizational unit (OU)** in the directory for the machine account.
- *ComputerAccountString*: A UTF-8 string that contains the value stored in the sAMAccountName attribute of the computer object in the domain database.

- *DNSComputerNameString*: A UTF-8 string that contains the Internet host name of the computer.
- *Spn1*: A UTF-8 string that contains a DNS-based service principal name (SPN) for the computer joining the domain.
- *Spn2*: A UTF-8 string that contains a NetBIOS-based SPN for the computer joining the domain.

The following statements define the sequence of message-processing operations:

1. If the **NETSETUP_MACHINE_PWD_PASSED** bit is set in *Options*, and the **NETSETUP_JOIN_UNSECURE** bit is not set in *Options*, the server MUST return ERROR_INVALID_PARAMETER. Otherwise, message processing continues.
2. If the **NETSETUP_MACHINE_PWD_PASSED** bit is set in *Options*, and *AccountName* is not NULL, the server MUST return ERROR_INVALID_PARAMETER. Otherwise, message processing continues.
3. If the **NETSETUP_MACHINE_PWD_PASSED** bit is set in *Options*, and either *Password* is NULL or the length of the *PasswordString* is zero, the server MUST return ERROR_PASSWORD_RESTRICTION. Otherwise, message processing continues.
4. If the **NETSETUP_MACHINE_PWD_PASSED** bit is set in *Options*, the value of *PasswordString* MUST be copied to the value of *ComputerPasswordString*, and *PasswordString* MUST be set to NULL.
5. If the **NETSETUP_JOIN_READONLY** bit is set in *Options*, and **NETSETUP_MACHINE_PWD_PASSED** bit is not set in *Options*, the server MUST return ERROR_INVALID_PARAMETER. Otherwise, message processing continues.
6. If the **NETSETUP_JOIN_READONLY** bit is set in *Options*, and the **NETSETUP_ACCT_CREATE** bit is set in *Options*, the server MUST return ERROR_INVALID_PARAMETER. Otherwise, message processing continues.
7. If the **NETSETUP_JOIN_READONLY** bit is set in *Options*, the server MUST perform all subsequent message processing as if **NETSETUP_DEFER_SPN_SET** and **NETSETUP_JOIN_UNSECURE** bits are set in *Options.*
8. If the server processing the message is already joined to a domain, and the **NETSETUP_DOMAIN_JOIN_IF_JOINED** bit is not set in *Options*, the server MUST return NERR_SetupAlreadyJoined. Otherwise, message processing continues.
9. If *DomainNameString* contains the character "\", *DomainNameString* MUST be truncated such that the value of *DomainNameString* is equal to the substring of *DomainNameString* that ends prior to the first "\" character, and *DomainControllerString* MUST be equal to the substring beginning after the first "\" character. This is the name of the target domain controller as specified by the caller.

   The specified domain controller MUST be validated by invoking the **DsrGetDcNameEx2** method ([MS-NRPC] section 3.5.4.3.1) on the DomainControllerString computer, specifying the following parameters:

   - *ComputerName* = *DomainControllerString*
   - *AccountName* = NULL
   - *AllowableAccountControlBits* = 0
   - *DomainName* = *DomainNameString*
   - *SiteName* = 0
   - *Flags* : if **NETSETUP_JOIN_READONLY** bit is set in *Options*, set *Flags* = (B | R); otherwise set *Flags* to (B | J | R)

If the call succeeds and DomainControllerInfo->DomainControllerName matches DomainControllerString, execution continues at step 8.

If the call fails, or the returned domain controller name does not match DomainControllerString, the server MUST invoke the **DsrGetDcNameEx2** method ([MS-NRPC] section 3.5.4.3.1) on the DomainControllerString computer, specifying the following parameters:

- *ComputerName* = *DomainControllerString*
- *AccountName* = NULL
- *AllowableAccountControlBits* = 0
- *DomainName* = *DomainNameString*
- *SiteName* = 0
- *Flags* : if **NETSETUP_JOIN_READONLY** bit is set in *Options*, set *Flags* = (B | S); otherwise set *Flags* to (B | J | S)

If the call fails, the server MUST stop message processing and return ERROR_NO_SUCH_DOMAIN. If the call succeeds and DomainControllerInfo->DomainControllerName matches DomainControllerString, execution continues at step 8. Otherwise, the server MUST stop message processing and return ERROR_INVALID_DOMAIN_ROLE.

10. If *DomainControllerString* was not initialized in the preceding step, the server MUST locate a domain controller for the domain specified in *DomainNameString*, and *DomainControllerString* MUST be set to the string name of the located domain controller. The same parameter values that are shown above are used except that the *ComputerName* parameter is set to NULL.
11. The **SiteName** ADM element SHOULD be updated with the client site name information that was returned as part of the call to **DsrGetDcNameEx2**.
12. *DomainNameString* MUST be a validated **domain name**. The validation process is specified in section 3.2.4.16, where *NameType* is **NetSetupDomain** from the **NETSETUP_NAME_TYPE** (section 2.2.3.2) enumeration. If this validation fails, the server MUST stop message processing and return the error specified in the validation process.
13. If **ComputerNameNetBIOS** is identical to *DomainNameString*, the server MUST return ERROR_INVALID_DOMAINNAME. Otherwise, message processing continues.
14. If the NETSETUP_MACHINE_PWD_PASSED bit is set in *Options*, the server MUST attempt to establish an **authenticated SMB** session with the domain controller named by the value of *DomainControllerString*. The client identity and authorization information that were used when establishing the SMB session are retrieved from **RPC** ([MS-RPCE] section 2.2.1.1.10 and [MS-RPCE] section 3.3.3.4.3).
15. If the NETSETUP_MACHINE_PWD_PASSED bit is set in *Options*, and the session fails to be established in the previous step with a non-authentication failure, the server MUST stop message processing and return the error. If the session fails to be established for some other reason, the server MUST attempt to establish an **anonymous session**. If an error occurs, the server MUST stop message processing and return that error. Otherwise, message processing continues.
16. If the NETSETUP_MACHINE_PWD_PASSED bit is not set in *Options*, the server MUST establish an authenticated SMB session with the domain controller named by the value of *DomainControllerString*. The credentials that are supplied during authentication are those specified in *PasswordString*, and the **security context** that is established MUST be that of *AccountName*. If an error occurs, the server MUST stop message processing and return that error. Otherwise, message processing continues.

17. The SMB session that was established in the previous steps and the security context associated with it MUST be used for any higher-layer RPC calls made to the domain controller over the SMB NCACN_NP protocol sequence ([MS-RPCE] section 2.1.1.2 and [MS-SMB] section 3.2.4.2.4).

18. The server MUST query the domain controller for its domain name and **SID** ([MS-LSAD] section 3.1.4.4.3).<71>

19. The server MUST store the values queried in the previous step in the local **DomainName** and **DomainSid** elements, as defined in section 3.2.1.6.

20. If the NETSETUP_MACHINE_PWD_PASSED bit is not set in *Options*, and either the NETSETUP_WIN9X_UPGRADE bit or the NETSETUP_JOIN_UNSECURE bit is set in *Options*, *ComputerPasswordString* is the first 14 characters of **ComputerName.NetBIOS** in lowercase.

21. If the NETSETUP_MACHINE_PWD_PASSED bit is not set in *Options*, and neither the NETSETUP_WIN9X_UPGRADE bit nor the NETSETUP_JOIN_UNSECURE bit is set in *Options*, *ComputerPasswordString* MUST be an ASCII string of randomly chosen characters. Each character's ASCII code MUST be between 32 and 122 inclusive. When randomly generating a password string, the server MUST generate 120 characters. Each character SHOULD be generated using the algorithm specified in [FIPS186-2] Appendix 3.1 and [RFC4086].<72>

22. The server MUST store the value of *ComputerPasswordString* locally for consumption by security-provider services when authenticating the computer. The stored password MUST be maintained by the Netlogon Protocol [MS-NRPC] in the **Password** ADM element, as defined in section 3.2.1.6.

23. If the value of the *MachineAccountOU* parameter is not NULL, the value of *MachineAccountOUString* MUST equal *MachineAccountOU*. If the value of *MachineAccountOU* is NULL, *MachineAccountOUString* MUST equal the value specified by the well-known object identified by the **GUID** with value GUID_COMPUTERS_CONTAINER_W ([MS-ADTS] section 6.1.1.4).

24. If the [RFC1777]-format name of the organizational unit (OU) where the object exists, as specified by the value of *MachineAccountOUString*, cannot be found in the domain database, the server MUST return ERROR_FILE_NOT_FOUND. Otherwise, message processing continues.

25. *ComputerAccountString* MUST be set to the UTF-8 string consisting of **ComputerNameNetBIOS** suffixed with a "$" character.

26. *DNSComputerNameString* MUST equal the UTF-8 string **ComputerNameFQDN**.

27. *Spn1* MUST be a UTF-8 string equal to the concatenation of "HOST/" and the value of *DNSComputerNameString*.

28. *Spn2* MUST be a UTF-8 string equal to the concatenation of "HOST/" and the value of *ComputerAccountString*.

29. If the NETSETUP_ACCT_CREATE bit is set in *Options*, the server MUST create the **domain object** in the domain *DomainNameParam* at *DomainControllerString*. Manipulation of the domain computer object state is exposed through LDAP protocols ([RFC2252], [RFC2253], and [MS-SAMR]). If the domain object already exists in an organizational unit (OU) ([MS-ADSC] section 2.217) that is different from the one specified in *MachineAccountOU*, the server MUST stop message processing and return NERR_UserExists. If the domain object already exists but the *MachineAccountOU* is NULL or refers to the organizational unit (OU) of the domain object, the server MUST return NERR_Success. Otherwise, message processing continues.

30. If the NETSETUP_ACCT_CREATE bit is not set in *Options* and the domain object does not already exist in the domain *DomainNameParam* at the domain controller, the server MUST stop message processing and return ERROR_NONE_MAPPED. Otherwise, message processing continues.

31. If the NETSETUP_ACCT_CREATE bit is not set in *Options*, and either the NETSETUP_WIN9X_UPGRADE bit or the NETSETUP_JOIN_UNSECURE bit is set in *Options*, the

server MUST send a request to the Netlogon Remote Protocol on the local computer to perform Netlogon authentication with the domain controllers. This is to validate that the value of *ComputerPasswordString* persisted locally equals the value of the password on the domain object in the **LDAP** attribute unicodePwd. If the authentication fails, the server MUST stop message processing and return ERROR_LOGON_FAILURE. Otherwise, message processing continues. For more information about Netlogon authentication between domain-joined computers and domain controllers, see [MS-NRPC].

32. If the NETSETUP_JOIN_READONLY bit is not set in *Options*, the following LDAP attributes on *DomainObject* MUST be set to the values shown in the table. The security context provided to the LDAP protocol is *AccountName* and the credential is *PasswordString*. For details about attributes and attribute names, see [MS-ADTS]. For details about LDAP, see [RFC2252] and [RFC2253].

LDAP attribute name	Value
userAccountControl ([MS-ADA3] section 2.342)	The USER_WORKSTATION_TRUST_ACCOUNT bit is set and the USER_ACCOUNT_DISABLED bit is not set. See the userAccountControl mapping table ([MS-SAMR] section 3.1.5.14.2) for details about the mapping of these bits in the LDAP protocol.
sAMAccountName ([MS-ADA3] section 2.222)	The value of ComputerAccountString.
unicodePwd ([MS-ADA3] section 2.332)	The value of ComputerPasswordString. Protocols that expose this attribute persist the **NT hash** of the ComputerPasswordString ([MS-SAMR] section 3.1.5.10).

33. The following LDAP attributes on *DomainObject* MUST be set to the values shown in the table unless the NETSETUP_DEFER_SPN_SET bit is set in *Options*.

LDAP attribute name	Value
dNSHostName ([MS-ADA1] section 2.185)	The value of *DNSComputerNameString*.
servicePrincipalName ([MS-ADA3] section 2.253)	Two values: *Spn1* *Spn2*

34. The server MUST configure the local Netlogon Remote Protocol [MS-NRPC] so that it is aware of being joined to a domain with the name *DomainNameParam*.

35. The server MUST configure the local **Windows Time Service (W32Time)** [WTSREF] so that it is aware of being joined to a domain.

36. The server SHOULD store the value *DNSComputerNameString* locally so that the DNS service registers name records for the local computer [NIS].<73>

37. The server SHOULD add the Domain Admins group to the local administrators group and the Domain Users group to the local users groups ([MS-SAMR] section 3.1.4.2).

38. The server MUST apply all state changes specified in section 3.2.4.13.2.

39. The server MUST stop impersonating the client by invoking the **StopImpersonatingClient** task (section 3.2.4.22.7).

If no errors occur, the server MUST return NERR_Success.

<122> Section 3.2.4.13: Windows NT, Windows 2000, Windows XP, Windows Server 2003, and Windows Server 2003 R2 do not implement this option.