

4.1.19.2 Server Behavior of the IDL_DRSReplicaAdd Method

Informative summary of behavior: The server adds a value to the [repsFrom](#) of the specified NC replica. If `ulOptions` contains `DRS_ASYNC_OP`, the server processes the request asynchronously. The client can be an administrative client or another DC. The client includes `DRS_WRIT_REP` in `ulOptions` if the specified NC replica is writable at the server. The client includes `DRS_NONGC_RO_REP` and `DRS_SPECIAL_SECRET_PROCESSING` in `ulOptions` if the specified NC replica is a read-only full replica on a read-only DC. The server adds a value to [repsFrom](#), and the value has `replicaFlags` derived from `ulOptions` (see below), `serverAddress` equal to `pszSourceDsaAddress` (`pszDsaSrc` if V1), and `schedule` equal to `rtSchedule`. If `ulOptions` contains `DRS_ASYNC_REP` but not `DRS_MAIL_REP` or `DRS_NEVER_NOTIFY`, the server sends a request to the DC specified by `pszSourceDsaAddress` to add a value to the [repsTo](#) of the specified NC replica by calling `IDL_DRSUpdateRefs`. Finally, the server begins a replication cycle by sending an `IDL_DRSGetNCChanges` request.

```
ULONG
IDL_DRSReplicaAdd(
    [in, ref] DRS_HANDLE hDrs,
    [in] DWORD dwVersion,
    [in, ref, switch_is(dwVersion)] DRS_MSG_REPADD *pmsgAdd);

options: DRS_OPTIONS
nc: DSName
partitionsObj: DSName
cr: DSName
rf: RepsFrom
msgIn: DRS_MSG_REPADD_V2
updRefs: DRS_MSG_UPDREFS /* See IDL_DRSUpdateRefs structures. */

ValidateDRSInput(hDrs, 5)

/* Validate the version */
if dwVersion != 1 and dwVersion != 2 then
    return ERROR_DS_DRA_INVALID_PARAMETER
endif
if dwVersion = 1 then
    msgIn := pmsgAdd^.V1
    msgIn.pszSourceDsaAddress = pmsgAdd^.V1.pszDsaSrc
else
    msgIn := pmsgAdd^.V2
endif

if msgIn.pNC = null
    or msgIn.pszSourceDsaAddress = null
    or msgIn.pszSourceDsaAddress = "" then
    return ERROR_DS_DRA_INVALID_PARAMETER
endif

options := msgIn.ulOptions
nc := msgIn.pNC^

partitionsObj :=
    select one o from children ConfigNC() where o!name = "Partitions"
cr := select o from children partitionsObj where o!nCName = nc
if cr = null then
    return ERROR_DS_DRA_BAD_NC
endif

if options - {DRS_ASYNC_OP, DRS_CRITICAL_ONLY, DRS_ASYNC_REP,
    DRS_WRIT_REP, DRS_INIT_SYNC, DRS_PER_SYNC, DRS_MAIL_REP,
    DRS_NONGC_RO_REP, DRS_SPECIAL_SECRET_PROCESSING, DRS_DISABLE_AUTO_SYNC,
    DRS_DISABLE_PERIODIC_SYNC, DRS_USE_COMPRESSION, DRS_NEVER_NOTIFY,
```

```

        DRS_TWOWAY_SYNC} # {} then
        return ERROR_DS_DRA_INVALID_PARAMETER
    endif

    if AmIRODC() and DRS_WRIT_REP in options then
        return ERROR_DS_DRA_INVALID_PARAMETER
    endif

    if AmIRODC() and DRS_MAIL_REP in options then
        return ERROR_DS_DRA_INVALID_PARAMETER
    endif

    if ObjExists(nc) then
        if not AccessCheckCAR(nc, DS-Replication-Manage-Topology) then
            return ERROR_DS_DRA_ACCESS_DENIED
        endif
    else
        if not AccessCheckCAR(DefaultNC(), DS-Replication-Manage-Topology)
            then
            return ERROR_DS_DRA_ACCESS_DENIED
        endif
    endif

    if DRS_ASYNC_OP in options then
        Asynchronous Processing: Initiate a logical thread of control
        to process the remainder of this request asynchronously
        return 0
    endif

    if ObjExists(nc) then
        if (IT_WRITE in nc!instanceType) # (DRS_WRIT_REP in options) then
            return ERROR_DS_DRA_BAD_INSTANCE_TYPE
        endif
        /* Disallow addition if server already replicates from this
        * source */
        if (select one v from nc!repsFrom
            where v.serverAddress = msgIn.pszSourceDsaAddress) # null
            then
            return ERROR_DS_DRA_DN_EXISTS
        endif
    endif

    if DRS_ASYNC_REP in options then
        if msgIn.pSourceDsaDN = null
            or not ObjExists(msgIn.pSourceDsaDN^)
            then
            return ERROR_DS_DRA_INVALID_PARAMETER
        endif
    endif

    if DRS_MAIL_REP in options then
        if msgIn.pTransportDN = null
            or not ObjExists(msgIn.pTransportDN^)
            then
            return ERROR_DS_DRA_INVALID_PARAMETER
        endif
    endif

    /* Construct RepsFrom value. */
    if msgIn.pSourceDsaDN # null then
        rf.uuidDsa := msgIn.pSourceDsaDN^!objectGUID
    endif
    if msgIn.pTransportDN # null then

```

```

    rf.uuidTransportObj := msgIn.pTransportDN^!objectGUID
endif
rf.replicaFlags := msgIn.ulOptions ∩ {DRS_DISABLE_AUTO_SYNC,
    DRS_DISABLE_PERIODIC_SYNC, DRS_INIT_SYNC, DRS_MAIL_REP,
    DRS_NEVER_NOTIFY, DRS_PER_SYNC, DRS_TWOWAY_SYNC,
    DRS_USE_COMPRESSION, DRS_WRIT_REP, DRS_NONGC_RO_REP,
    DRS_SPECIAL_SECRET_PROCESSING }
rf.schedule := msgIn.rtSchedule^
rf.serverAddress := msgIn.pszSourceDsaAddress^
rf.timeLastAttempt := current time

nc!repsFrom := nc!repsFrom + {rf}
if msgIn.ulOptions ∩ {DRS_ASYNC_REP, DRS_NEVER_NOTIFY, DRS_MAIL_REP}
    = {DRS_ASYNC_REP} then
    /* Enable replication notifications by requesting the server DC
    * to add a repsTo for this DC. */
    updRefs.pNC^ := ADR(nc)
    updRefs.pszDsaDest := NetworkAddress of this DC
    updRefs.uuidDsaDest := dc.serverGuid
    updRefs.ulOptions := {DRS_ASYNC_OP, DRS_ADD_REF, DRS_DEL_REF}
    if DRS_WRIT_REP in msgIn.ulOptions then
        updRefs.ulOptions := updRefs.ulOptions + {DRS_WRIT_REP}
    endif
    Send updRefs request by calling IDL_DRSUpdateRefs() on server
    msgIn.pszSourceDsaAddress^
endif

Perform a replication cycle as a client of IDL_DRSGetNCChanges. Call
ReplicateNCRequestMsg (section 4.1.10.4.1) to form the first
request and send it. If not DRS_MAIL_REP in msgIn.ulOptions, then
wait for the response, process it (section 4.1.10.6), send the next
request, etc. until the replication cycle is complete. If there are
any failures from this replication attempt, err should be assigned
an appropriate error value.

return err

```

5.39 DRS_OPTIONS

DRS_OPTIONS is a concrete type for a set of options sent to and received from various drsuapi methods.

This type is declared as follows:

```
typedef unsigned long DRS_OPTIONS;
```

Five elements of the set are interpreted differently by different methods; such elements have multiple symbolic names.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
A	G	A	A	W	I	P	M	A	T	C	G	G	N	S	F	F	F	S	S	R	N	S	I	P	S	D	D	U	N	S	G
S	C	R	L	R	S	S	R	S	S	O	A	S	R	N	S	S	S	Q	U	G	S	S	S	E	F	A	P	C	N	P	P
	/		L					R				/	R	/	/	I	P			/	Y				S	S					
	U		/					/				L	R	N						N											
	N		D	R				E				O	F	S						D											

X: Unused. MUST be zero and ignored.

AS (DRS_ASYNC_OP): Perform the operation asynchronously.

GC (DRS_GETCHG_CHECK): Treat ERROR_DS_DRA_REF_NOT_FOUND and ERROR_DS_DRA_REF_ALREADY_EXISTS as success for calls to [IDL_DRSUpdateRefs \(section 4.1.26\)](#).

UN (DRS_UPDATE_NOTIFICATION): Identifies a call to [IDL_DRSReplicaSync](#) that was generated due to a replication notification. See [\[MS-ADTS\]](#) section 3.1.1.5.1.5 for more information on replication notifications. This flag is ignored by the server.

AR (DRS_ADD_REF): Register a client DC for notifications of updates to the NC replica.

ALL (DRS_SYNC_ALL): Replicate from all server DCs.

DR (DRS_DEL_REF): Deregister a client DC from notifications of updates to the NC replica.

WR (DRS_WRIT_REP): Replicate a writable replica, not a read-only partial replica or read-only full replica.

IS (DRS_INIT_SYNC): Perform replication at startup.

PS (DRS_PER_SYNC): Perform replication periodically.

MR (DRS_MAIL_REP): Perform replication using SMTP as a transport.

ASR (DRS_ASYNC_REP): Populate the NC replica asynchronously.

IE (DRS_IGNORE_ERROR): Ignore errors.

TS (DRS_TWOWAY_SYNC): Inform the server DC to replicate from the client DC.

CO (DRS_CRITICAL_ONLY): Replicate only system-critical objects.

GA (DRS_GET_ANC): Include updates to ancestor objects before updates to their descendants.

GS (DRS_GET_NC_SIZE): Get the approximate size of the server NC replica.

LO (DRS_LOCAL_ONLY): Perform the operation locally without contacting any other DC.

NRR (DRS_NONGC_RO_REP): Replicate a read-only full replica. Not a writable or partial replica.

SN (DRS_SYNC_BYNAME): Choose the source server by network name.

RF (DRS_REF_OK): Allow the NC replica to be removed even if other DCs use this DC as a replication server DC.

FS (DRS_FULL_SYNC_NOW): Replicate all updates in the replication cycle, even those that would normally be filtered.

NS (DRS_NO_SOURCE): The NC replica has no server DCs.

FSI (DRS_FULL_SYNC_IN_PROGRESS): When the flag `DRS_FULL_SYNC_NOW` is received in a call to `IDL_DRSReplicaSync`, the flag `DRS_FULL_SYNC_IN_PROGRESS` is sent in the associated calls to [IDL_DRSGetNCChanges](#) until the replication cycle completes. This flag is ignored by the server.

FSP (DRS_FULL_SYNC_PACKET): Replicate all updates in the replication request, even those that would normally be filtered.

SQ (DRS_SYNC_REQUEUE): This flag is specific to the Microsoft client implementation of `IDL_DRSGetNCChanges`. It is used to identify whether the call was placed in the replicationQueue more than once due to implementation-specific errors. This flag is ignored by the server.

SU (DRS_SYNC_URGENT): Perform the requested replication immediately; do not wait for any timeouts or delays. For information about urgent replication, see [\[MS-ADTS\]](#) section 3.1.1.5.1.6.

RG (DRS_REF_GCSPN): Requests that the server add an entry to `repsTo` for the client on the root object of the NC replica that is being replicated. When `repsTo` is set using this flag, the notifying client DC contacts the server DC using the service principal name that begins with "GC" (section [2.2.3.2](#)).

ND (DRS_NO_DISCARD): This flag is specific to the Microsoft implementation. It identifies when the client DC should call the requested `IDL_DRSReplicaSync` method individually, without overlapping other outstanding calls to `IDL_DRSReplicaSync`. This flag is ignored by the server.

NSY (DRS_NEVER_SYNCED): There is no successfully completed replication from this source server.

SS (DRS_SPECIAL_SECRET_PROCESSING): Do not replicate attribute values of attributes that contain secret data.

ISN (DRS_INIT_SYNC_NOW): Perform initial replication now.

PE (DRS_PREEMPTED): The replication attempt is preempted by a higher priority replication request.

SF (DRS_SYNC_FORCED): Force replication, even if the replication system is otherwise disabled.

DAS (DRS_DISABLE_AUTO_SYNC): Disable replication induced by update notifications.

DPS (DRS_DISABLE_PERIODIC_SYNC): Disable periodic replication.

UC (DRS_USE_COMPRESSION): Compress response messages.

NN (DRS_NEVER_NOTIFY): Do not send update notifications.

SP (DRS_SYNC_PAS): Expand the partial attribute set of the partial replica.

GP (DRS_GET_ALL_GROUP_MEMBERSHIP): Replicate all kinds of group membership. If this flag is not present nonuniversal group membership will not be replicated.

For information about the Windows versions in which these flags were introduced, see the following behavior note. [<37>](#)