Exchange 2000 Public Folder Replication

Version 2.0

Introduction

This document explains in detail the Exchange 2000 Public Folder replication process. In the past, there has been little documentation on how this process works. The document bridges the gap between the low level MDB source code documentation and the high level help supplied with Exchange 2000 Server.

The replication engine in Exchange 2000 works in a similar way as to the replication engine in Exchange 5.5. Much of what is documented here can equally be applied to previous versions of Exchange.

This document cannot answer every question on Public Folder replication, nor can it provide details on all the possible replication scenarios. Instead it describes the replication process, what settings are important and how public folders interact with the Active Directory and email in general. From a troubleshooting perspective, knowing *how* something is supposed to work makes it much easier to figure out *why* something is not working. This is what this document aims to do.

The chapters have been written to be as "stand alone" as possible. However, to avoid duplication this was not always possible. You are advised to read through the whole document, as some subjects (especially permissions) are covered in multiple places.

The document is broken down into several main sections, covering the basics of Public Folders, an overview of the **replication process**, details about the different types of **replication messages**, plus many examples of the process in action and how this process scales in larger topologies. It also covers public folder **directory entries, emailing to public folders, permissions, transport and referrals**. While these latter issues are not directly related to public folder replication, they touch on it so are included here. Also there are deployment issues with the placing of Public Stores. **Finally there are sections on common problems, how to troubleshoot them and some tips picked up by the Exchange 2000 PFREPL test team during Public Folder testing.**

Who this document is aimed at

PSS Support Engineers, Microsoft Consulting Services, deployment specialists, experienced IT administrators, experienced Exchange 5.5 administrators.

What this document assumes some knowledge of

Administering Exchange 2000 or Exchange 5.5, Windows 2000 Active Directory, using LDP or ADSI Edit, using the Event Viewer, basic mail transport, and administering Public Folders.

Public Folder Replication Basics

This section provides a high level overview of Public Folders and replication. It also explains some terms used later on in the documentation.

Public Folder Overview

Top Level Hierarchy

A Top Level Hierarchy (TLH) is the root of a public folder tree. In Exchange 5.5 there was only one TLH called "Public Folders". In Exchange 2000 there can be several. The "Public Folder" TLH is just one of many Public Folder trees. It is commonly known as the MAPI TLH and performs exactly the same tasks as it did in Exchange 5.5 (and will replicate with the Exchange 5.5 Public Folder tree). However, in Exchange 2000, there can also be multiple additional trees, commonly known as Application TLHs (App TLH).

Each TLH has a directory entry, which, among other things, contains a Backlink to the Directory Names (DNs) of all the stores in the TLH.

The MAPI TLH will be secured in the directory under the first administrative group in the organization.

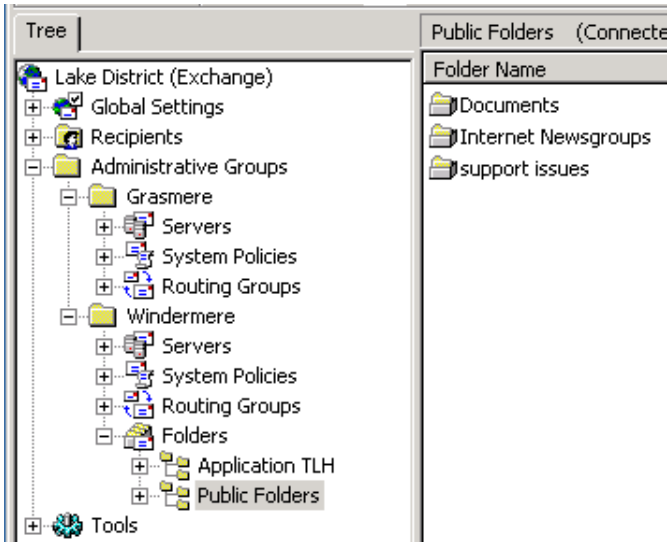
Example

```
CN=Public Folders,CN=Folder Hierarchies,CN=Windermere,CN=Administrative Groups,CN=Lake
District,CN=Microsoft Exchange,CN=Services,CN=Configuration,DC=cumbria,DC=extest,DC=microsoft,DC=com
```

Tip

Use MMC snap-ins to create a new console just for viewing the Folders container. This saves having to search for the Folders container.

Additional Folders containers can be created in other Admin Groups, and TLHs can be moved between them.



Virtual Directories

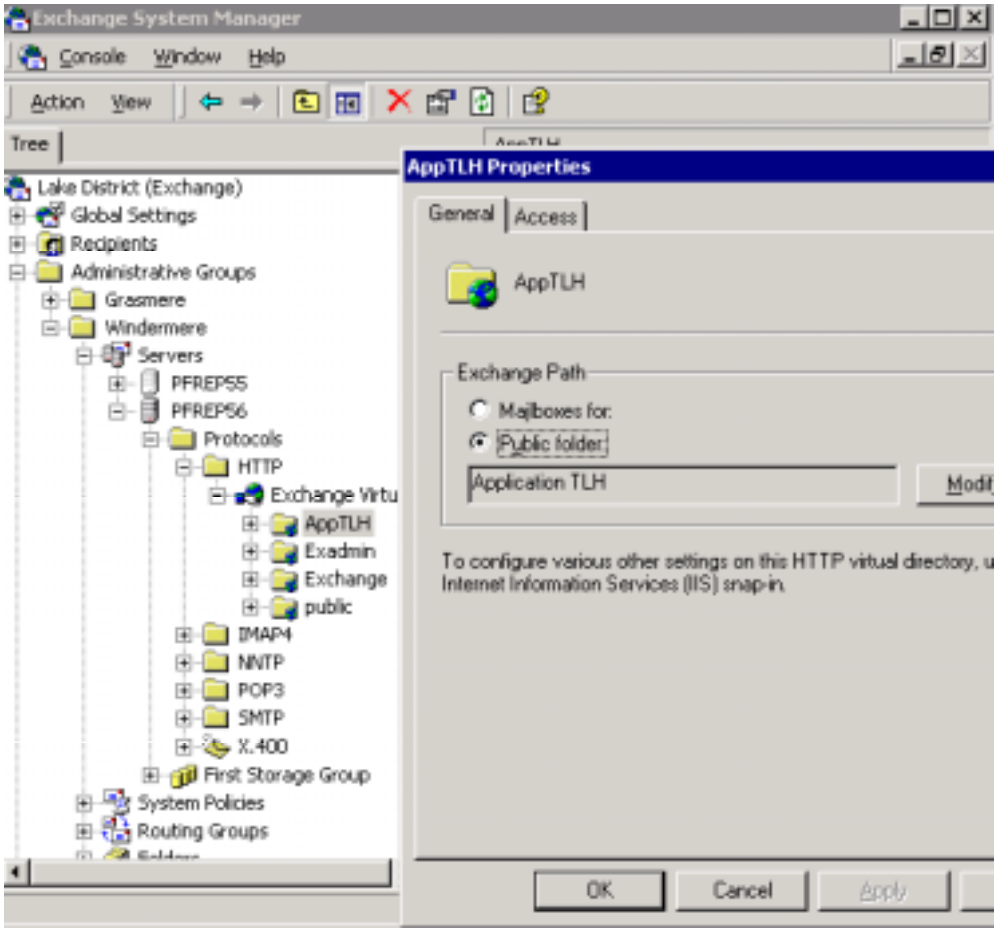
To allow Outlook Web Access (OWA) via Http to a public folder, there must be a virtual directory for the TLH on the server the client is accessing.

MAPI TLH virtual roots are created automatically, and are called “public”

Therefore, http://<servername>/public will access the MAPI TLH public store.

When additional TLHs are created, servers that contain stores in the TLHs can have virtual directories created for them.

Example



It is possible to create virtual directories on one server that point to other servers for the TLH. This requires additional configuration through IIS Admin.

Public Folder Database

Public Folders are stored in a Public Folder Database. In Exchange 5.5 the Public Folder Database was stored in the pub.mdb file (and the Information Store transaction logs). In Exchange 2000 the default Public Folder database (MAPI TLH) is contained in pub*x*.edb & pub*x*.stm (where x is a number), and is created automatically on server installation.

Additional Public Folder databases (stores) can be created to store folders from other Public Folder hierarchies (App TLHs).

Configuring Multiple Public Stores

- There can only be one hierarchy per store.
- A server can have multiple Public Folder Stores.
- A server cannot have multiple stores containing the same hierarchy. A new store can only be created if a hierarchy exists which is not currently assigned to a store on the server.
- There can only be one MAPI TLH in the Organization.

What this means in practice

By default only the MAPI TLH exists. To create additional Public Stores, you must first create a new App TLH. Once you've created another TLH you can then create a new store and assign the TLH to that store.

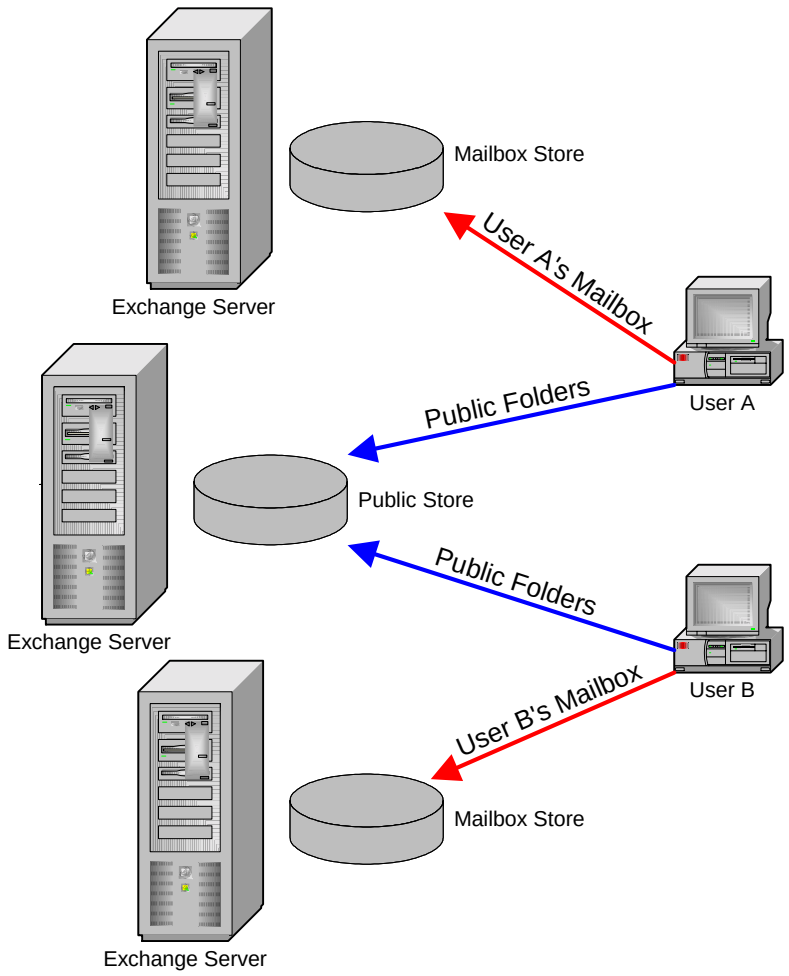
Public Folder Server

In Admin Groups (or Exchange 5.5 sites) containing more than 3 servers, it is usual to deploy specific Public Folder Servers. This significantly reduces replication traffic and makes administration of Public Folders much easier. The Mailbox Servers have had their Public Stores removed, and the Public Folder servers have few or no users on them (or have even had the Mailbox Store(s) removed).

Explanation

Users A & B have their mailboxes on different servers.

However, they both access the same server for public folders



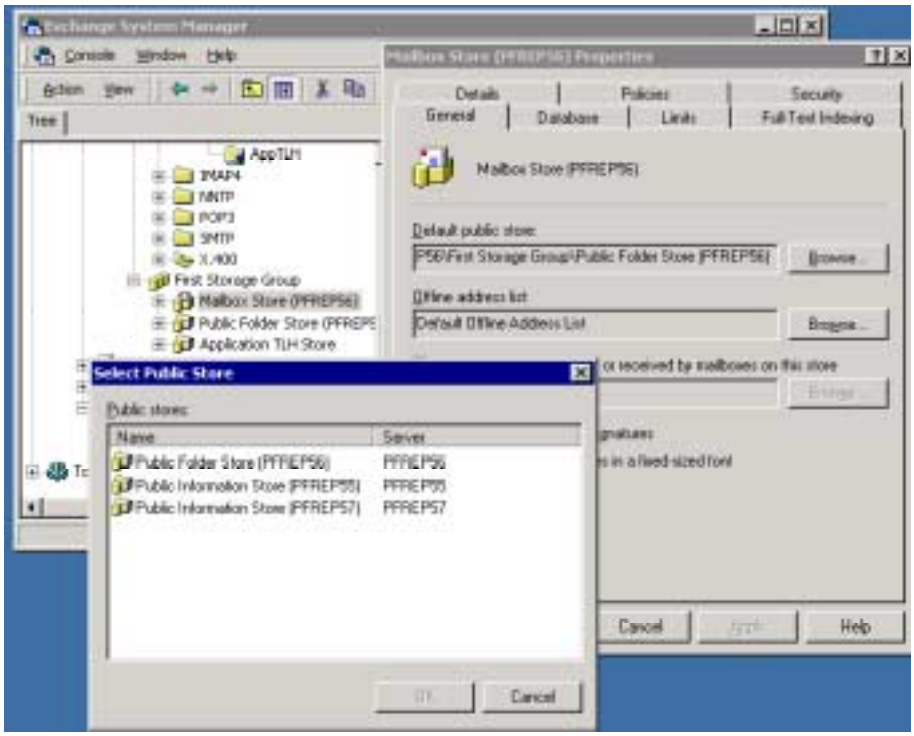
Tip

If the server is not going to contain replicas of public folders, remove the public stores to reduce unnecessary hierarchy replication messages. See **Replication Status** for further information

Mailbox stores are then pointed at the Public Folder Servers for their default Public Folder Store.

Mailboxes Properties

Changing the mailboxes' default public folder stores



IPM & Non-IPM_Subtree

The public folder database is divided into two trees. The IPM_Subtree and the non-IPM_Subtree

Switching Between IPM & Non IPM_Subtree (or System Folders)

Right Click on the TLH object in ESM and toggle between "View System Folders" and "View Public Folders"

IPM_Subtree (Public Folders)

This contains the folders visible to users and clients. For example a folder created by Microsoft Outlook will exist in the IPM Subtree. Folders in the IPM_Subtree can be accessed directly by clients, searched and used to store user data.

Non IPM_Subtree (System Folders)

This contains folders not directly accessible by users. The folders in this tree replicate in an exactly the same way as IPM_Subtree Folders, but cannot be manipulated directly by users.

Some examples of folders in the non-IPM_Subtree:

- Site Folders (Free & Busy data, Events registry, MAPI Forms, Offline Address List)
- Restrictions*
- Views*

*Not replicated

Site folders are visible when viewing "System Folders". They replicate just like ordinary folders and their replica lists can be modified in exactly the same way as non-system folders.

First Server in Admin Group

The first server in an Admin Group will hold copies of Offline Address Lists, Free & Busy data and replicas of other Site Folders. The location of these folders can be changed through ESM.

Each Admin Group has a Site Folder Server, which is the first server in the site. This determines which server is responsible for ensuring Site Folders exist. It is an attribute of the Admin Groups directory entry

Example

```
1> siteFolderGUID: <ldp: Binary blob>;

1> siteFolderServer: CN=Public Information Store (PFREP60),CN=First
Storage
Group,CN=InformationStore,CN=PFREP60,CN=Servers,CN=Mercury,CN=Administr
ative Groups,CN=Solar System,CN=Microsoft
Exchange,CN=Services,CN=Configuration,DC=berks,DC=extest,DC=microsoft,D
C=com;
```

Deleting Public Folder Stores

There are several ways that a Public Store can be removed. This will briefly look at some of the ways and any problems they present.

Deleting a Public Store in ESM

This is the cleanest way to remove a public folder store. Before removing the store any folders that exist on this server should be moved (or at least replicated) to another server, because the contents of any public folders that are only replicated to this public store will be permanently lost once its deleted.

In ESM **right click** on the store and choose **delete**.

A warning will be displayed:

```
"It is strongly recommended that any public folders replicas are
removed from this public folder store. The contents of any public
folders that are only replicated to this public folder store will
be permanently lost.  Continue (Y/N)?"
```

If the public store is used as the default public folder store by mailboxes, there will be a prompt to choose an alternate public folder store for the mailboxes.

If the public store is used by one or more Offline Address Lists, there will be a prompt to choose an alternate public folder store for the Offline Address Lists. If there is no other Exchange 2000 store that can be used to house the Offline Address Lists, the store cannot be deleted.

Deleting Public Store Database

If for some reason the public folder database (e.g. pubx.edb) is deleted, a new one will be created when the store remounts. The hierarchy will backfill and if any of the folders on the deleted store had replicas on other servers the content will backfill as well.

If this store contained Site Folders (e.g. Free & Busy) and they were not replicated anywhere else, it may be necessary to recreate the site folders by running Guidgen.exe.

If the store contained Offline Address Lists, it will be necessary to **Rebuild** the Offline Address Lists.

Uninstalling a Public Folder Server

Exchange 2000 servers should be removed from the Organization by running Setup and selecting uninstall. This will clean up the directory as the server is being removed. You will not be allowed to uninstall a server until certain tasks have been completed (e.g. change Offline Address List server etc.)

You cannot uninstall a server that is running an SRS.

Removing a Public Folder Server.

This is the most destructive way of removing a server and can cause the most problems. Selecting **Server → Remove Server** is a way of forcing the Server out of the Organization. It bypasses all the checks made by the other methods.

The only time this should be used is if the actual server itself has been lost (e.g catastrophic failure and no backup). Even then it should be used with caution.

If the server removed was a Site Folder Server, then Guidgen.exe will have to be used to select a new Site Folder Server.

If the server removed was an Offline Address List server, then a new server will have to be chosen, and the Offline Address Lists rebuilt.

If the server contained an SRS, then the ADC's CAs may have to be changed.

If the server contained an SRS, a re-arbitration by the Super KCC may occur which can cause major problems to Exchange 5.5 servers. For more information on this see **Replication Problems**.

Using Guidgen.exe

For information on how to reset site folders see **Q152960**. At the time of writing this article has not been updated for Exchange 2000. Follow the instructions, but instead of modifying the attributes in the Exchange 5.5 DS Raw Mode, use ADSI Edit to change the **siteFolderGUID** & **siteFolderServer** attributes on the appropriate Admin Group object in the Windows 2000 Active Directory. The store must be remounted to pick up the changes.

Replicas and Ghosted folders

The TLH hierarchy is replicated to all the stores assigned to the TLH. This is the representation of the folders as seen by the Exchange System Manager (ESM) and clients. However the content only exists in actual *replicas* of the folders. Folders that exist only in the hierarchy on a server and don't have a local replica are called *ghosted* folders.

Note

They still have an entry in the folders table, and most of the property tags, but do not contain any content. Basically their Folder Table rows don't have an associated MsgFolder Table

A note about the Hierarchy

The hierarchy is actually the content of a special folder, and this folder is replicated to all stores in the TLH. The hierarchy is the content of folder 1-1. Therefore hierarchy replication is the replication of the content of folder 1-1.

Client Access & Referral

Different Clients can access different TLHs

Client	MAPI TLH	App TLH
MAPI (Outlook)	Yes	No
IMAP4*	Yes	No
POP3	No	No
HTTP-DAV (Outlook Web Access)	Yes	Yes
IFS	Yes	Yes

When a client accesses a public folder from the hierarchy the store will compute the nearest replica that contains the content of that folder, and then refer the client to that store. If the replica is not on the client's local Public Folder server, the client will make a new connection to that server and access the content.

Note

IFS does not support referral. You cannot view ghosted folders via IFS.

Also the Microsoft IMAP4 client does not support folder referral (but other IMAP clients may).

Further Information

For more information on the referral mechanism see **Public Folder Referral and Public Folder Affinity**

Mail Enabled Folders

A Mail Enabled Folder is a public folder that has a directory entry, so that it can be looked up in the address book and emailed.

In Exchange 5.5 all Public Folders were mail enabled (by default their directory entries were hidden and created in the Recipients container).

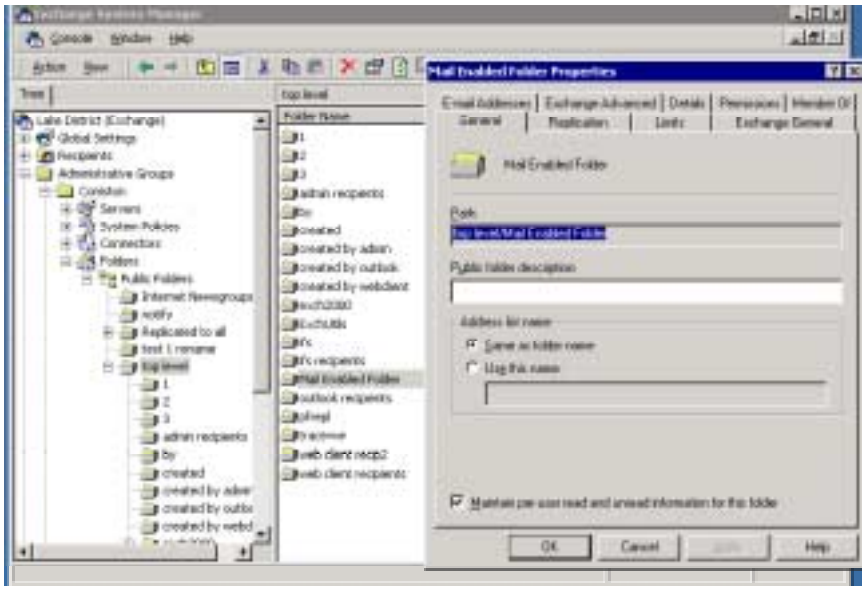
In Exchange 2000, folders can be mail enabled or mail disabled depending on whether the Exchange Organization is in mixed mode or native mode. Below is a summary of the possible settings.

TLH	Mixed Mode	Native Mode
MAPI TLH	Always mail enabled By default hidden from GAL.	Either mail enabled or disabled, default is disabled.
App TLH	Either mail enabled or disabled, default is disabled. If mail enabled, by default they are visible in GAL.	Either mail enabled or disabled, default is disabled. If mail enabled, by default they are visible in GAL.

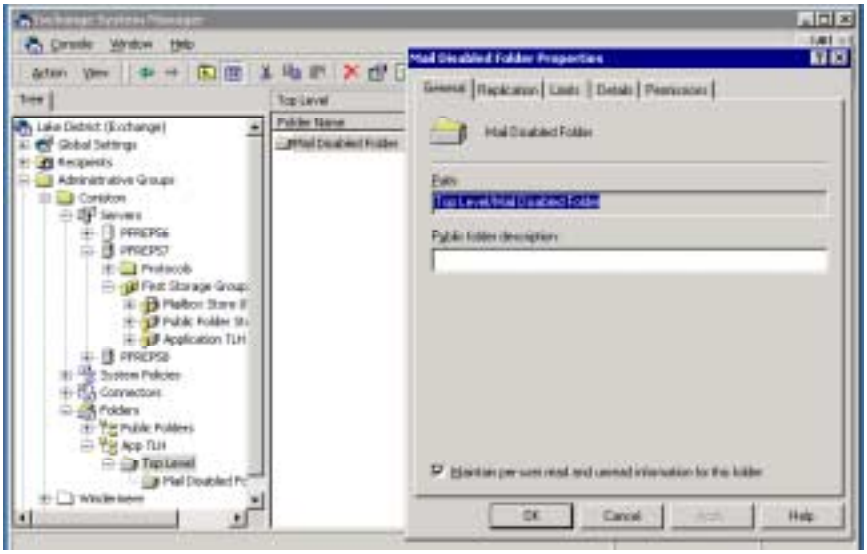
Note

MAPI folders are always mail enabled in mixed mode. This is for backwards compatibility with Exchange 5.5. The Exchange 5.5 Admin program expects to find a directory entry with a public folder, and without one you cannot administer the folder from Exchange 5.5.

Mail Enabled Folder Properties



Mail Disabled Folder Properties



The mail-disabled folder does not have any email properties.

Tip

The option to Mail Enable a folder is always available on a MAPI folder in Mixed Mode. This is so that you can re-mail enable the folder (i.e. recreate the directory entry) if it gets deleted for any reason.

Recipient Update Service

The Recipient Update Service (RUS) is controlled by the system attendant and runs on at least one server in the Organization. It is responsible for adding mail attributes to objects in the Windows 2000 Active Directory (W2K AD).

As public stores replicate updates by emailing each other, public stores must have mail attributes (mail, proxyAddresses etc.). It is the responsibility of the RUS to stamp these attributes on the public stores' directory objects. For more information see **Public Store Directory Entries**.

Clusters

There can only be one public store for each TLH per cluster. This is to prevent problems if the cluster fails over to another server. If each node had a database belonging to the same TLH, when the cluster failed over, multiple databases for the same TLH would exist on the same server, which is not allowed.

Replication

Public Folder replication is the transmittal of the data stored in public folders between stores in the same TLH, via an email based replication engine. The process is exactly the same for MAPI and App TLHs. The folder hierarchy is replicated via *hierarchy replication* messages (replication of the content of Folder ID 1-1) and the content of folders is replicated via *content replication* messages between replicas of individual folders. In addition to this there are *Backfill* replication messages, *Status* messages and *Status Request* messages, which keep replication between stores synchronized.

Note

FID is Folder ID. Internally the store addresses folders by a FID which is a hex id e.g. 1-2A45. A FID is a row in the Folders Table in the store. Similarly Messages are referenced by MIDs (Message IDs), which is a row in the MsgFolder Table.

Replication makes use of standard transports to send email to other stores. If an update has to go to multiple stores, then a single replication message is generated, addressed to the multiple stores (in other words the replica list for the folder – in the case of the hierarchy, this is all the stores in the TLH). It is up to transport & routing to decide how the message needs to be split up. It is exactly the same is if a user adds multiple recipients to the TO: line of a message.

Mail based

Public Folder replication is mail based. Replication messages are email messages sent between the Public Stores in each TLH. This means that there must be an email path between the stores for replication to work (see **The Replication Process** & **Transport and Routing**)

Replication Messages

Replication Messages	
Transport independent	Replication messages can be sent over different types of email link.
System Messages	Replication messages are treated as system messages. This means that they do not obey normal restrictions applied to user email messages (in Exchange 5.5 directory replication messages were also system messages), such as size and delivery restrictions.
Addressed to other Public Folder Stores	Replication messages are sent by a **store to other stores**. The receiving store then updates the folders based on the information contained in the replication message. The individual folders' directory entries are **not** used for folder replication. They are purely used to allow clients to email the folders.

Public Store Directory Entries

Folders replicate by sending email between information stores. This means that Public Folder Stores require email addresses (added by RUS). Below is an example of a MAPI Public Store's directory entry.

Some DS attributes have been removed for clarity (e.g USN number, when changed etc.)

```
>> Dn: CN=Public Folder Store (PFREP57),CN=First Storage
Group,CN=InformationStore,CN=PFREP57,CN=Servers,CN=Coniston,CN=Administrative
Groups,CN=Lake District,CN=Microsoft
Exchange,CN=Services,CN=Configuration,DC=cumbria,DC=extest,DC=microsoft,DC=com
       1> msExchOwningPFTree: CN=Public Folders,CN=Folder
Hierarchies,CN=Coniston,CN=Administrative Groups,CN=Lake District,CN=Microsoft
Exchange,CN=Services,CN=Configuration,DC=cumbria,DC=extest,DC=microsoft,DC=com;
       1> homeMDBBL: CN=SMTP (PFREP57-{409AD800-749B-414E-A980-
2B551268854C}),CN=Connections,CN=Lake District,CN=Microsoft
Exchange,CN=Services,CN=Configuration,DC=cumbria,DC=extest,DC=microsoft,DC=com;
       1> adminDisplayName: Public Folder Store (PFREP57);
       1> cn: Public Folder Store (PFREP57);
       1> displayName: Public Folder Store (PFREP57);
       1> mail: PFREP57-IS@Coniston.LakeDistrict.com;
       1> legacyExchangeDN: /O=Lake
District/OU=Coniston/cn=Configuration/cn=Servers/cn=PFREP57/cn=Microsoft Public MDB;
       1> distinguishedName: CN=Public Folder Store (PFREP57),CN=First Storage
Group,CN=InformationStore,CN=PFREP57,CN=Servers,CN=Coniston,CN=Administrative
Groups,CN=Lake District,CN=Microsoft
Exchange,CN=Services,CN=Configuration,DC=cumbria,DC=extest,DC=microsoft,DC=com;
       1> objectCategory: CN=ms-Exch-Public-
MDB,CN=Schema,CN=Configuration,DC=cumbria,DC=extest,DC=microsoft,DC=com;
       3> objectClass: top; msExchMDB; msExchPublicMDB;
       1> objectGUID: 409ad800-749b-414e-a980-2b551268854c;
       2> proxyAddresses: SMTP:PFREP57-IS@Coniston.LakeDistrict.com; X400:c=US;a=
;p=Lake District;o=Coniston;s=PFREP57-IS;;
       1> name: Public Folder Store (PFREP57);
       1> showInAdvancedViewOnly: TRUE;
       1> textEncodedORAddress: c=US;a= ;p=Lake District;o=Coniston;s=PFREP57-IS;;
       1> activationSchedule: <ldp: Binary blob>;
       1> activationStyle: 1;
       1> homeMTA: CN=Microsoft
MTA,CN=PFREP57,CN=Servers,CN=Coniston,CN=Administrative Groups,CN=Lake
District,CN=Microsoft
Exchange,CN=Services,CN=Configuration,DC=cumbria,DC=extest,DC=microsoft,DC=com;
       1> mailNickname: PFREP57-IS;
       1> deliveryMechanism: 1;
       1> msExchEDBFile: E:\Program Files\Exchsrvr\mdbdata\pub1.edb;
       1> msExchEDBOffline: FALSE;
       1> maximumObjectID: <ldp: Binary blob>;
       1> msExchOwningServer: CN=PFREP57,CN=Servers,CN=Coniston,CN=Administrative
Groups,CN=Lake District,CN=Microsoft
Exchange,CN=Services,CN=Configuration,DC=cumbria,DC=extest,DC=microsoft,DC=com;
       1> msExchPollInterval: 15;
       1> quotaNotificationSchedule: <ldp: Binary blob>;
       1> quotaNotificationStyle: 1;
       1> msExchReplicationMsgSize: 300;
       1> msExchReplicationSchedule: <ldp: Binary blob>;
       1> msExchReplicationStyle: 2;
       1> msExchSLVFile: E:\Program Files\Exchsrvr\mdbdata\pub1.stm;
       1> msExchMaxCachedViews: 11;
       1> msExchPoliciesIncluded: {CB137506-78E1-4583-BB76-9F69D57DFAAE},{26491CFC-
9E50-4857-861B-0CB8DF22B5D7};
       1> msExchDatabaseCreated: TRUE;
       1> msExchInconsistentState: 4;
```

App TLH Stores' Directory Entries

The directory entry for a store assigned to an App TLH is essentially the same. One attribute to note, however, is the LegacyExchangeDN.

No matter what the store is called, its LegacyExchangeDN will always be of the form:

```
/O=<org>/OU=<Admin Group>/cn=Configuration/cn=Servers/cn=<server
name>/cn=MICROSOFT PUBLIC MDB<+ 8 digit random number>
```

Example

```
legacyExchangeDN: /O=Lake
District/OU=Coniston/cn=Configuration/cn=Servers/cn=PFREP57/cn=MICR
OSOFT PUBLIC MDB36595809
```

Further Information

Historical Note

The reason for this is that the PF replication engine in the past used the DN on a message to determine whether the message was a replication message, email to a folder, or an email incorrectly addressed to the store. If the replication message (or indeed an email message being sent to a folder on that store) is to be delivered successfully the string "MICROSOFT PUBLIC MDB" must be contained in the recipient DN address.

Replicating APP TLHs in mixed mode Organization is covered in more detail later.

Implications for IMC

This has implications for folder replication via an Exchange 5.5 IMC. The IMC in Exchange 5.5 does not resolve addresses in a message to directory entries by default. So names in the P2 are not resolved to DNs. This will prevent MAPI TLH replication working over an IMC. You need to allow the IMC to resolve names by setting the IMC registry key ResolveP2 = 1. This applies equally to Exchange 5.5 ⇔ Exchange 5.5 replication as it does to Exchange 2000 ⇔ Exchange 5.5 replication. For further details on setting ResolveP2 see ***Q174755****. To allow App TLH replication via an IMC an additional step is required, see* ****Special Instructions for App TLH replication over Exchange 5.5 IMC***

Packing & Unpacking

The process of putting the data into the replication message ready to be sent out is called *Packing*. The process of retrieving the replication data from the replication message is called *Unpacking*.

Multiple hierarchy updates and content updates *for the same folder* can be packed into a single replication message. This reduces mail traffic as a single message can contain multiple updates (reduces overhead of P1 & P2 headers). Hierarchy updates cannot be packed into the same replication message as content updates.

Example

Explanation

This single content replication message contains three content updates. In this case three items (post1, post2 & post3 were added to the folder.

```
An incoming replication message was processed.

 Type: 0x4
 Message ID: 9-5F2E
 Folder: (9-4A4C) IPM_SUBTREE\Documents\Social

 Database "First Storage Group\Public Folder Store (PFREP56)".
 CN min: a-13B7
 CN max: a-13B9
 MIDs: 3 1: a-11B7, a-13B7
--- : post1 : 8/20/2000 11:57:16 PM
2: a-11B8, a-13B8
--- : post2 : 8/20/2000 11:57:20 PM
3: a-11B9, a-13B9
--- : post3 : 8/20/2000 11:57:24 PM

 MIDSET deleted: {0}

 Server: /O=LAKE
DISTRICT/OU=GRASMERE/CN=CONFIGURATION/CN=SERVERS/CN=PFREP57/CN=MIC
ROSOFT PUBLIC MDB
```

Change Numbers

All updates (create, delete & modify) are assigned Change Numbers (CNs). These are used by the replication engine to track updates. Every modification to a folder is given a Change Number. When a folder replicates an update to another server the CNs are included with the update. The CNs are then used by the receiving server to determine whether this is a new change, and also whether it is missing any data. A set of CNs is called a CNSet.

More Information

CNs are similar to Update Sequence Numbers (USNs) used in Directory Replication. However, Public Folder Replication is very different from Directory Replication, so this is where the similarities end.

InterOrg Replication

The Exchange 2000 replication engine can only replicate folders within the same Exchange Organization (exactly the same as Exchange 5.5). To replicate folders between Organizations there is a tool provided with Exchange 2000 called the InterOrg Replication Connector (Exchsync).

It can be found in the

`\Support\Exchsync`

directory on the Exchange 2000 CD.

It consists of two programs:

- Configuration Utility – exscfg.exe
- Replication Utility – exssrv.exe

These programs are not covered by this document. For more information see the instructions accompanying the utilities.

Summary

This section has covered the basics of Public Folder replication and defined terms used in public folder replication.

- Exchange 2000 supports multiple TLHs.
- Only one MAPI TLH can exist in the hierarchy.
- The MAPI TLH can replicate with Exchange 5.5.
- The hierarchy is replicated to all the stores assigned to that TLH.
- Folders that exist only in the hierarchy (i.e. contain no content) are ghosted folders.
- Replication occurs by sending email between stores.

The next section looks at the different types of replication messages.

Replication Message Types

There are 5 replication message types. The most common ones are hierarchy replication messages (remember this is effectively the content replication of FID 1-1) and content replication messages (replicating content between individual folder replicas).

Others are Backfill messages, Status Messages and Status Request messages. Status messages are used to check replicas are synchronized. If a store finds that it is not synchronized it will issue a Backfill request to another server to retrieve the missing content.

Tip

To capture replication message details in event viewer set Exchange Server diagnostics “Replication Incoming” & “Replication Outgoing” to maximum.

Hierarchy Replication Messages

A Hierarchy replication message is a replication message between replicas of FID 1-1. FID 1-1 will be replicated to all stores in the same TLH; so hierarchy messages will be broadcast to all stores in the TLH.

Hierarchy Replication Message	
Type	0x2
Purpose	Replicates Public Folder hierarchy between servers in the same TLH. Used whenever there is a change to the FID row in the Folder Table
Event 3018 Outbound Replication Message	An outgoing replication message was issued. Type: 0x2 Message ID: 1-47A3 Database "First Storage Group\Public Folder Store (PFREP61)" CN min: 1-479E, CN max: 1-47A0 RFIs: 1 1: 1-4275,1-1,28 IPM_SUBTREE\Andy's RC top level replicated to all IDCN Deleted: {0}
Event ID 3028 Inbound Replication Message	An incoming replication message was processed. Type: 0x2 Message ID: 1-4283 Database "First Storage Group\Public Folder Store (PFREP65)". CN min: b-479E CN max: b-47A0 RFIs: 1 1: b-4275,1-1,28 IPM_SUBTREE\Andy's RC top level replicated to all IDCN deleted: {0} Server: /O=YORKS/OU=LEEDS/CN=CONFIGURATION/CN=SERVERS/CN=PFREP61/CN=MICROSOFT PUBLIC MDB
Comments	Example of changes which will generate a hierarchy replication are: Creating or deleting a folder. Modifying the folder (except it's contents) – e.g. renaming the folder, changing its replica list, display name, permissions and description. In fact any change other than actually adding content to the folder will be replicated by a hierarchy replication.

Content Replication Messages

Content replication messages replicate content updates between replicas of individual folders. A store will only send a content replication to another store that holds a replica of the folder.

Content Replication Message	
Type	0x4
Purpose	Replicates content between replicas of folders.
Event 3020 Outbound Replication Message	An outgoing replication message was issued. Type: 0x4 Message ID: 1-4FCE Folder: (1-4279) IPM_SUBTREE\Andy's RC top level replicated to all\PFREP61-pf1 Database "First Storage Group\Public Folder Store (PFREP61)". CN min: 1-1, CN max: 1-4DCC Message IDs: 1 1: 1-4BCC, 1-4DCC --- : post #1 : 4/7/2000 12:20:32 AM MIDSET Deleted: 1-1,1-4BCB
Event 3030 Inbound Replication Message	An incoming replication message was processed. Type: 0x4 Message ID: 1-46A9 Folder: (b-4279) IPM_SUBTREE\Andy's RC top level replicated to all\PFREP61-pf1 Database "First Storage Group\Public Folder Store (PFREP65)". CN min: b-1 CN max: b-4DCC MIDs: 1 1: b-4BCC, b-4DCC --- : post #1 : 4/7/2000 12:20:32 AM MIDSET deleted: b-1,b-4BCB Server: /O=YORKS/OU=LEEDS/CN=CONFIGURATION/CN=SERVERS/CN=PFREP61/CN=MICROSOFT PUBLIC MDB
Comments	Examples of changes that will generate a content replication message are: Posting items to a folder. Modifying items in a folder. Deleting items from a folder.

Backfill Replication Messages

Backfilling is the process by which stores that have missed replication updates can request a re-send of missing data. There are two parts to the Backfill process: Backfill Request and Backfill Response. In order for a store to issue a backfill request, it must "discover" that it is not synchronized, by detecting a gap in a folder's CNSet. This is accomplished either through normal replication, or from Status Messages sent by other stores.

Backfill Request

Backfill Request Message	
Type	0x8
Purpose	To Request a backfill of missing CN sets for a particular folder.
Event 3014 Outbound Replication Message	An outgoing replication message was issued. Type 0x8 Message ID: 1-4499 Database "Storage Group 2\TLH on 65 and 61". CNSET: 4-366B,4-366E CNSET(FAI): {0} Server: /O=YORKS/OU=LEEDS/CN=CONFIGURATION/CN=SERVERS/CN=PFREP61/CN=MICROSOFT PUBLIC MDB38128847
Event 3024 Inbound Replication Message	An incoming replication message was processed. Type: 0x8 Message ID: 1-3676 Database "Storage Group 2\TLH on 61 and 65". CNSET: 1-366B,1-366E CNSET(FAI): {0} Server: /O=YORKS/OU=HUDDERSFIELD/CN=CONFIGURATION/CN=SERVERS/CN=PFREP65/CN=MICROSOFT PUBLIC MDB01528527
Comments	The examples here are for a Hierarchy Backfill request (FID 1-1); exactly the same principles apply to a content Backfill request (see later for example of this).

Backfill Response

Backfill Response Message	
Type	0x80000002 (Hierarchy FID 1-1) or 0x80000004 (content replica)
Purpose	Response to a Backfill Request, containing the requested data
Event 3019 Outbound Replication Message	An outgoing replication message was issued. Type: 0x80000002 Message ID: 1-3678 Database "Storage Group 2\TLH on 61 and 65". CNSET: 1-366B,1-366E CNSET(FAI): {0} RFIs: 1 1: 1-2338,1-1,28 IPM_SUBTREE\backfill 2 IDCN Deleted: {0}
Event 3029 Inbound Replication Message	An incoming replication message was processed Type: 0x80000002 Message ID: 1-449B Database "Storage Group 2\TLH on 65 and 61". CNSET: 4-366B,4-366E CNSET(FAI): {0} RFIs: 1 1: 4-2338,1-1,28 IPM_SUBTREE\backfill 2 IDCN deleted: {0}
Comments	The above is the hierarchy Backfill response for the previous backfill request.

Status Messages

Status messages are sent by one store to another store, to allow the receiving store to determine whether it is synchronized with the sender.

Status Message	
Type	0x10
Purpose	Sends details about the current state (CN sets) of a folder to a store that contains a replica of that folder.

Event 3017 Outbound Replication Message	`An outgoing replication message was issued.`
	
	`Outgoing message type 0x10`
	`Message ID: 1-4764`
	`Folder(s): (1-1) IPM_SUBTREE`
	
	`Database "Storage Group 2\TLH on 61 and 65".`

Event 3027 Inbound Replication Message	`An incoming replication message was processed.`
	
	`Type: 0x10`
	`Message ID: 1-44C7`
	`Database "Storage Group 2\TLH on 65 and 61".`
	`Folder(s): (1-1) IPM_SUBTREE`

Comments	The event does not log the actual CNSets that are included in the Status Message.

Status Request Messages

A Status Request Message is sent by a store to another store in order trigger replication of missing updates. Status Requests are less common in Exchange 2000 than they were in Exchange 5.5. They occur when a new replica of a folder is created on a store. In other words the store "knows" that it's bound to be missing data, because a newly created replica has just been placed on the store so it will have to backfill. A hierarchy status request is generated when a new store is created.

These two occasions when Status Requests are generated are actually very similar. A new replica of a folder will generate a Status Request for the content; a new store will generate a Status Request for the hierarchy – because a new hierarchy folder has just been created.

Additionally Status Requests are used when folder replicas are removed from stores. For more information see **Modifying the Replica List**.

Note

In Exchange 5.5 a Status Request for all replicas (including the hierarchy) was generated whenever the Information Store was restarted. It was possible to disable this via a registry key. In Exchange 2000 it has been determined that this generated too much replication traffic, especially with multiple databases and backup systems that require a shutdown of the Information Store, so it has been removed.

Status Request Message	
Type	0x20
Purpose	To request CNSets when a folder's replica list changes (e.g. a folder being replicated to a new server), or to prompt a remote store into sending a Status Message (in the case replicas being removed)..
Event 3017 Outbound Replication Message	An outgoing replication message was issued. Message ID: 1-53B7 Folder(s): (1-2333) IPM_SUBTREE\test 2 Database "Storage Group 2\TLH on 61 and 65".
Event 3027 Inbound Replication Message	An incoming replication message was processed. Type: 0x20 Message ID: 1-451C Database "Storage Group 2\TLH on 65 and 61". Folder(s): (4-2333) IPM_SUBTREE\test 2 Server: /O=YORKS/OU=LEEDS/CN=CONFIGURATION/CN=SERVERS/CN=PFREP61/CN=MICROSOFT PUBLIC MDB38128847
Comments	The above example is a Status request for Folder "test 2" or FID 1-2333.

Summary

This section has covered the 5 types of replication message.

Replication Message	When Used
Hierarchy (0x2)	Replicates hierarchy changes from public store to all other stores in the same TLH.
Content (0x4)	Replicates content changes from one replica to all other "content" replicas (i.e. non ghosted) of that folder.
Backfill (0x8)	Request missing data (in CNSets) from another store (both hierarchy and content).
Backfill Response (0x80000002 or 0x80000004)	Sends missing data to a store which requested missed updates (CNSets)
Status Message (0x10)	Sends the current CNSets of a folder to another replica(s) of that folder. Used for hierarchy (i.e. replicas of folder 1-1) and content (specific content replicas).
Status Request (0x20)	Requests CNSets to be replicated, or Status Messages to be returned. Used for hierarchy and content.

The next section will show replication messages in use and how they interact with one another.

The Replication Process

This part will look at how the replication process actually works.

To simplify matters the examples contain only two or three servers replicating with each other. The next section will go into further detail on what happens with multiple servers in multiple Sites/Admin Groups.

Public Stores send replication messages to each other via email. Therefore, there must be an email path between the stores for replication messages to be received.

Replication Thread

A thread runs continually in the store.exe process which polls for replication events. Replication events occur at specific time intervals. When this "timed event" occurs the replication thread spins off a new thread that performs the specified replication task.

For example, by default:

- Hierarchy replication events occur every 5 minutes
- Content replication events occur every 15 minutes.
- Status replication events occur every 24 hours

What this section covers

The following areas of replication will be covered:

Hierarchy Replication → replicating folder information.

Content Replication → replicating the content of folders to other replicas.

Replication Backfill → how the store can request missing updates.

Replication Status Messages → how the Public Stores remain synchronized.

Modifying the Replica List → what happens when folders are add or removed from Stores.

The Replication State Table → where all the data about replication CNs, backfill data and updates is stored.

Modifying the hierarchy

A hierarchy replication message is generated whenever the hierarchy is modified.

Examples of hierarchy modification

- Creating, deleting & renaming a folder.
- Modifying folder permissions.
- Modifying the description.
- Changing the replication schedule or priority.
- Almost any change made to a folder, except actually adding content, is a hierarchy change.
- Modifying replica lists (this will be looked at separately. See **Modifying the Replica List** for more information).

Remember

Hierarchy Replication is similar to content replication. The hierarchy is merely the content of folder 1-1, so all the rules that apply to content replication also apply to hierarchy replication.

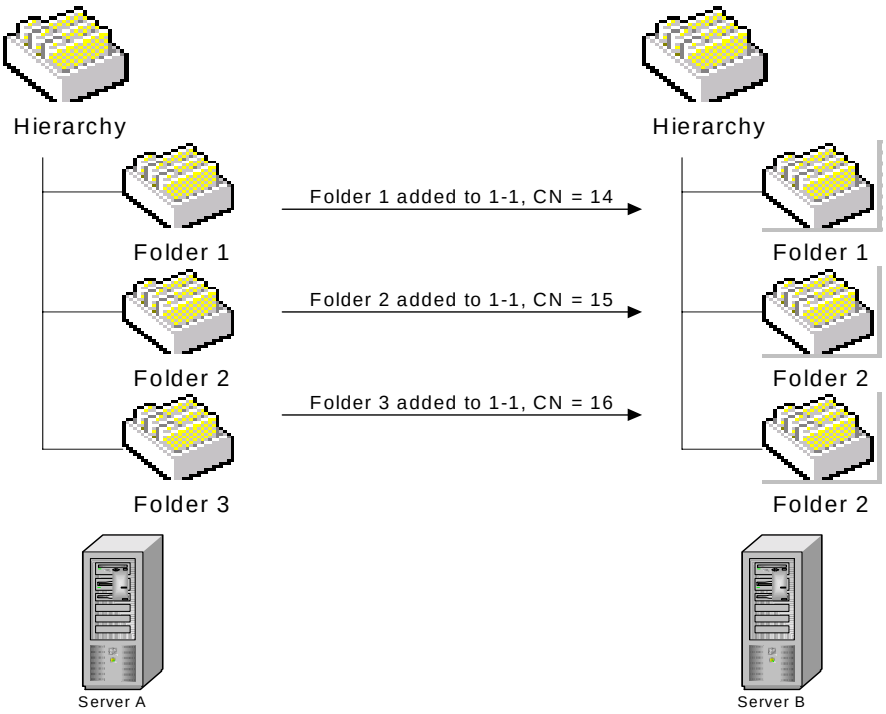
Example

Explanation

Folder 1 is added to Server A. Server A replicates the hierarchy changes to Server B.

Subsequent folders are added to Server A and these hierarchy changes also replicate to Server B.

The content replicas exist on Server A. Server B has ghosted folders.



Content Replication

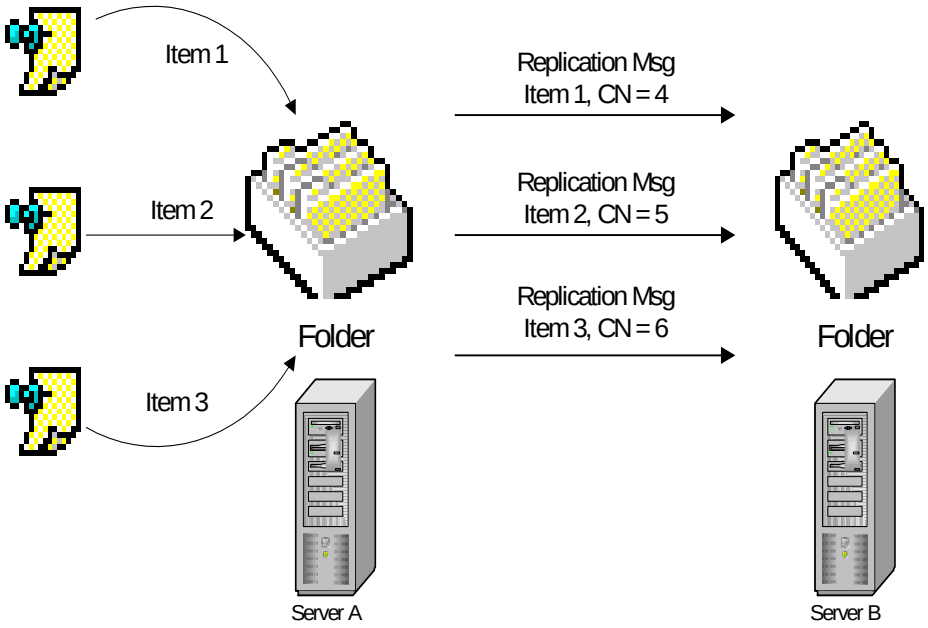
Folder contents replicate between individual replicas of folders. Whenever the contents of a folder are modified, these are tracked with CNs. When the replication interval is reached the changes are replicated to all other Public Stores that have a replica of the folder.

Example

Explanation

Item 1 is posted into a folder on Server A, which has a replica on Server B. The store on Server A replicates the change to the store on Server B.

Similarly, Items 2 and 3 are posted and replicated.



More Information

The above example shows 3 items being posted to a folder. Exactly the same process occurs for items by modified or deleted.

The Backfill Process

Folders remain synchronized via the backfill process. Folders will backfill only when they are missing contents. Therefore, for a folder to issue a backfill request, it must first "discover" that it has missed an update. This is accomplished by looking for a missing sequence in the folder CNSets for individual folders.

Both content and hierarchy backfill work in the same way. A hierarchy backfill is issued when there is a gap in the CNSets for folder 1-1, content backfills are requested for gaps in any other folder.

Backfill

For a store to issue backfill requests to retrieve missing updates, it must first realize it is out of sync. This is either achieved by subsequent replication updates, or from Status Messages.

The backfill process can take a long time – especially if a store is down and has missed the original replication update and the subsequent Status message (see **Complications and problems** later). It may not realize that it is missing content until further replication messages arrive.

Tip

If a folder is out of sync and does not seem to get back in sync after the normal backfill time-outs, modify a "correct" replica of a folder (e.g for hierarchy modify the hierarchy, for missing content modify the content). This will force a replication message to be sent to the out of sync store, and trigger a backfill request. See **Replication Problems** for more info.

Important Note

*In Exchange 2000 it is no longer safe to simply wipe queues as it was in Exchange 5.5 (e.g. MTACheck /rp). In order to reduce the amount of public folder replication traffic, the amount of Status Messages sent by stores has been significantly reduced. Therefore, if replication messages are deleted, folders can become unsynchronized and not realize it. See **Replication Status** for more information on this process.*

Backfill Array

The backfill array is used to store pending backfill requests. When the store "discovers" a folder is out of sync it writes an entry into the backfill array. This entry is a pending request for the missing data from another replica of the folder. The entry will stay in the backfill array until it times out, at which point a backfill request will be issued. The default backfill timeouts are given in the table below.

	Intra Site	Inter Site
Initial Backfill	6 hours	12 hours
First Backfill retry	12 hours	24 hours
Subsequent Backfill retries	24 hours	48 hours

If the first backfill attempt goes unanswered, then subsequent backfill attempts will wait longer before being sent.

The reason these times are so long is to prevent unnecessary backfilling. The replication message may be en route, delayed or stuck somewhere waiting for a connector's schedule. If the backfill timeout was too short, stores will start issuing backfill requests for messages already on the way.

Example

Backfill Process

This demonstrates how the backfill process works.

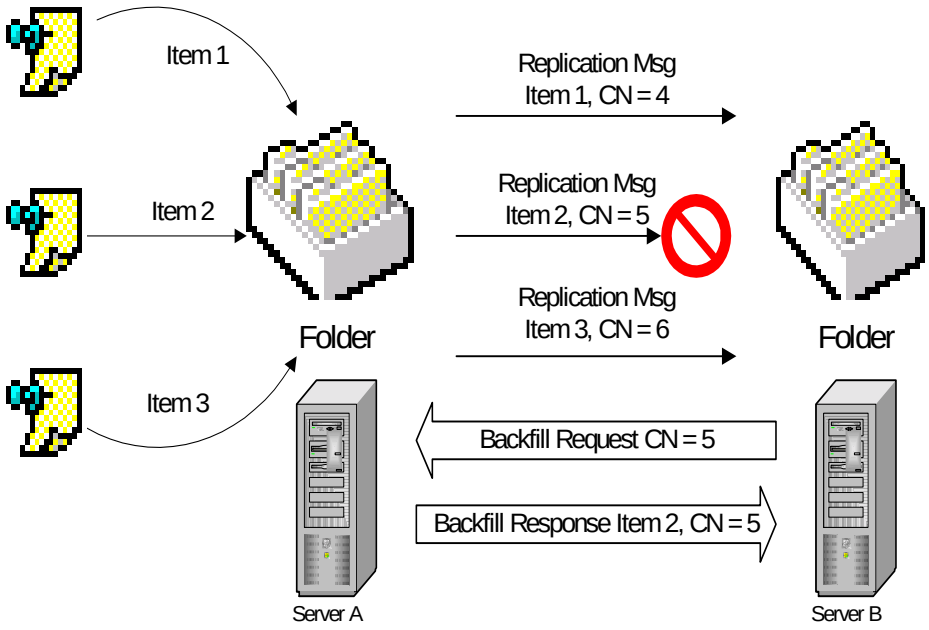
Item 1 is posted into the folder on Server A, and is replicated to Server B.

Item 2 is posted into the folder on Server A, however its replication message fails to be delivered to Server B

Item 3 is posted into the folder on Server A, and is replicated to Server B

Server B now knows that it is missing a change from Server A for the folder. An entry is written into the backfill array. When the backfill timeout is exceeded a backfill request is sent. Server A then generates a backfill response and replicates the missing data.

Below is an example of one method by which a store will generate a backfill request. A folder is replicated between two servers, Server A and Server B.



Note

If the missing update is received before the backfill request is sent, the entry for the backfill request is cleared from the backfill array.

Example

The following is the traced output from a folder's backfill array, and the subsequent Backfill Request and Response.

```
TAG 0:  ptagBackfill
TAG 0:  BACKFILL[0]
TAG 0:    FILETIME: 8/19/2000 11:43:16 PM (no expiration)
TAG 0:    FAI: FALSE
TAG 0:    FID: 9-4A4C
TAG 0:    cnMin: a-13B4
TAG 0:    cnMax: a-13B4
TAG 0:    Replid: 0
TAG 0:    Backfill ID: 9-5C43
TAG 0:    fInactive: fFalse
TAG 0:    ftTimeout: 8/20/2000 11:43:16 AM (38333 seconds)
TAG 0:  BACKFILL[1]
TAG 0:    FILETIME: 8/19/2000 11:43:16 PM (no expiration)
TAG 0:    FAI: TRUE
TAG 0:    FID: 9-4A4C
TAG 0:    cnMin: a-13B4
TAG 0:    cnMax: a-13B4
TAG 0:    Replid: 0
TAG 0:    Backfill ID: 9-5C43
TAG 0:    fInactive: fFalse
TAG 0:    ftTimeout: 8/20/2000 11:43:16 AM (38333 seconds)

----- Backfill request outgoing -----
TAG 0:  Replication Outstanding Backfill Limit: 50 (default value -
registry variable not found)
TAG 8e:  [Send Backfill Request] FID: 9-4A4C
TAG 8e:    cnsetBackfill:
TAG 0:  [a-13B4,a-13B4]
TAG 8e:    cnsetBackfillFAI:
TAG 0:  [a-13B4,a-13B4]
TAG 8e:    to MDB /o=Lake
District/ou=Grasmere/cn=Configuration/cn=Servers/cn=PFREP57/cn=Microso
ft Public MDB

----- Backfill response incoming -----
TAG 8e:  [Rcv Chg 1] FID: 9-4A4C, MID: a-11B4
TAG 8e:    from MDB /o=Lake
District/ou=Grasmere/cn=Configuration/cn=Servers/cn=PFREP57/cn=Microso
ft Public MDB
TAG 8e:  Folders: Insert locally owned CN range  (BACKFILL)9-4A4C: a-
13B4, a-13B4
```

Explanation

Folder 9-4A4C is missing one item (CNSet a-13B4).

The Backfill Array has been populated because the folder received a subsequent replication message.

The backfill timeout will expire in 38333 seconds (~10 hours).

When the backfill timeout expires, a backfill request is generated, asking for the missing CNSets (a-13B4), and sent to Server PFREP57.

PFREP57 responds with the missing CNSet, which in this case is a posting with a MID of "a-11B4".

Hierarchy Backfill

These examples were for backfilling missing content from a folder. Exactly the same process is used for missing hierarchy updates.

No Subsequent Replication Messages to indicate updates are missing

If there were no subsequent replication messages, how does the store "know" that it would be missing data? This is what Status Messages are for and are covered next.

Replication Status

Status messages fall in two categories, Status Requests, and Status Messages.

Status Messages

A Status message is sent from one store to another to indicate the current state of a particular folder on the sending server. If the Status message indicates that the sending store has more up to date information about the folder, then the receiving store will write an entry into its backfill array to request a backfill. If the CNSets are shown to be equal (or the receiving server is more recent) no action is taken.

A store will generate a Status message under two circumstances.

1. In response to a Status Request

If a store receives a Status request from another store requesting a Status Message. This will occur when the replica list of folders are being changed (specifically a folder being removed from a server). For more information see **Modifying the Replica List**.

2. 24 hours after the last local change to a folder

This is a significant change from previous versions of Exchange. When a change is made to a folder an expiry time for a Status message is set on that folder. If another change is made to that folder the expiry time is reset back to 24 hours. Once the expiry time is reached a Status message will be generated for that folder. Once this has occurred, the expiry time is cleared and no other status messages will be generated for that folder unless another change is made, which will reset the expiry time back to 24 hours.

Replication Status Thread

The thread which runs to check to see if a Status message should be sent only runs at 12:15 am & pm GMT by default.

When it runs, it checks to see if the timeout has been reached for any folders, in which case it will broadcast a Status Message.

Therefore, it could take up to 36 hours of zero changes, to generate a Status Message.

The replication status thread timing can be altered with the following registry settings:

Replication Send Status Timeout
Replication Send Status Alignment
Replication Send Status Alignment Skew

With the reduced number of Status Messages sent by Exchange 2000, it should not be necessary to modify the default values. More information on these settings can be found in **Q203170.**

Explanation

Item 1 is added to a folder on Server A.

The Status Message Timeout is set to 24 hours.

Item 2 is added to the folder 12 hours later. The Status Message Timeout is reset to 24 hours again.

No further changes are made, so when the timeout expires a Status Message is sent.

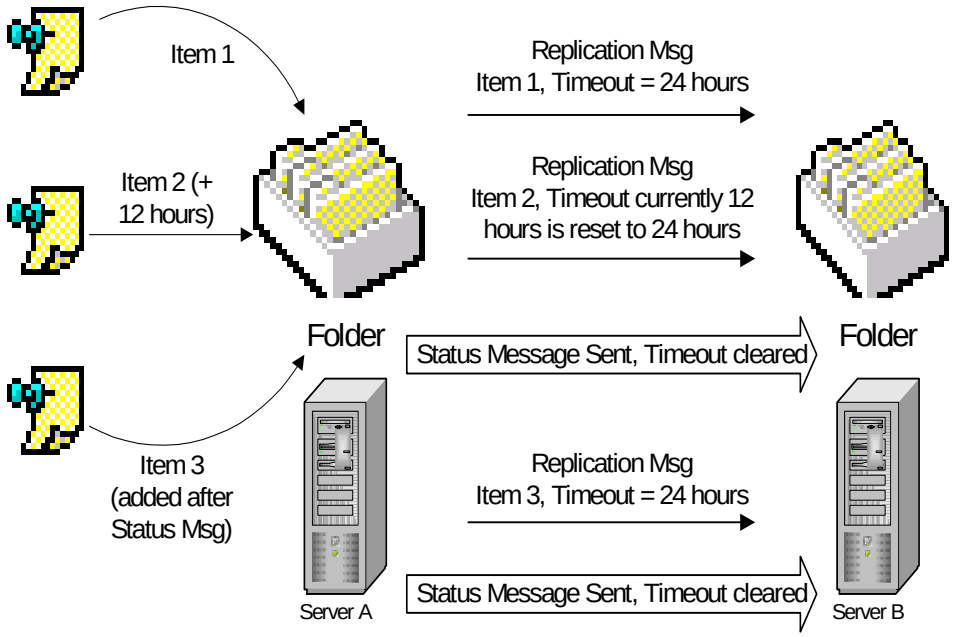
The timeout is not reset.

No further Status Messages will be sent for this folder.

Item 3 is then added to the folder. The Status Message Timeout is reset to 24 hours again.

If no further changes are made a Status message will be sent when this new timeout expires.

Example



Exchange 5.5

Exchange 5.5 did not use the same concept of a Status Timeout. A Status message would always be sent every 24 hours to all other replicas of the folder, regardless of whether any changes had been made to the folder. This meant that even folders that never changed continued to send Status Messages every 24 hours.

Impact on replication

Replicas that are modified often (i.e. at least once every 24 hours) will never send a Status message. They don't need to because they will be sending regular updates, so any dropped replication messages will soon be identified by the arrival of subsequent replication messages.

Replicas that are not modified will not constantly generate Status messages every 24 hours. If a folder is modified and then remains unchanged it will generate a Status message 24 hours later, then will not generate another Status message until another change is made.

Note

This greatly reduces the amount of background replication traffic in an organization. It was not uncommon in large organizations for Public Stores to generate 30Mb worth of background replication messages *each* and send then to each other. This could easily overwhelm slow links and cause backlogs of email traffic.

Status Message Triggers Backfill

If a Status Message is received that indicates a Public Store is missing data, the backfill array is populated with a backfill request. If the update is not replicated in the backfill timeout period, then a backfill request will be sent to request the missing data.

Example

This is an example of how a Status Message can trigger a backfill request

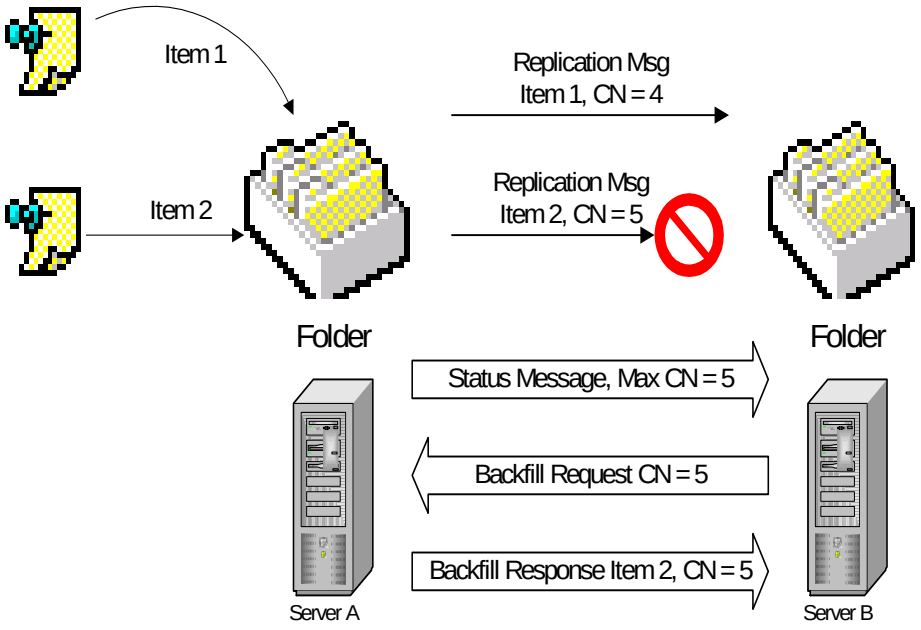
Explanation

Item 1 is posted into a folder on Server A and replicated to Server B

Item 2 is posted into the folder, but its replication message fails to be delivered to Server B.

No other updates are made the folder.

24 hours after the last change made to the folder, a Status Message is sent to Server B. Server B now realizes it is missing an update, and so writes an entry into the Backfill Array. Once the backfill timeout expires, a backfill request is sent, and the missing update is replicated back.



Note

If the missing update is received before the backfill request is sent, the entry for the backfill request is cleared from the backfill array.

Status Requests

A Status Request occurs when a store wishes to get a remote server's status for a particular folder. Depending on the circumstances a Status Request may trigger a Status message to be sent back.

A Status Request will be generated under the following circumstances

1. When a new Public Store is mounted for the first time

When a store is mounted for the first time, it will generate a Hierarchy Status Request for folder 1-1. (There can't be any content replicas assigned to this store so all it is missing will be the hierarchy). This will trigger another store into sending a Hierarchy replication message containing the missing updates.

2. When the replica list of a folder is changed

Whenever the replica list of a folder changes, a Status Request message is generated. Adding a new replica, deleting a replica or a Temp Replica Rehome will all generate Status Requests. As these are slightly more complex than a normal replication process they are discussed separately. See **Modifying the Replica List**

Note

A Temp Replica Rehome can occur when a client creates a folder. A client will always initially create a folder on its local Public Folder server. If the parent folder does not exist on this server, the newly created folder will be moved so that it inherits its parent's replica list. From the point of view of replication, this is just another replica list change. For more information see **Q253297**

Modifying the Replica List

Modifying the replica list is a hierarchy change. However, because the replica list is changing (folder replicas are either being created or removed from a server), Status Messages and Status Requests are also used.

Adding a new replica

When a new replica is added to a folder several steps occur:

1. A hierarchy replication message is sent out, to replicate the change in the folders replica list.

2. A server that has the content sends out a Status Request message to the Server with the new replica.

3. A content replication message is sent from the server with content to the server with the new replica

4. The server with the new replica also sends out a Status Request.

The steps may not always occur in the same order, depending on which store finds out about the replica change first. If the administrator makes the change on a server that has a content replica, then the steps happen in the above order. If the server which hosts the new replica finds out about the change first then it will issue a Status Request to a server which holds a replica.

Deleting a Replica

When a replica is removed from a server, the folder is not deleted immediately. Instead, it is put into a Delete Pending state. When a folder is in a Delete Pending state it cannot be viewed by a client or be administered (ESM will not show it on the list of folders hosted on the store).

It exists so that other replicas can still replicate any missing data from it.

Only when the Delete Pending Folder receives a status messages from another replica that the folders are in sync, will the deleted replica actually be removed.

The following steps occur:

1. The folder is removed from the replica list

2. A hierarchy message is replicated out indicating the change in the folder's status (active → delete pending)

3. The server that hosts the "delete pending" folder sends out a Status Request – which must be responded to.

4. A server with a replica responds to the Status Request with a Status Message. If the Status Message indicates that CNSets are at least as current as the replica being deleted, the store will proceed to the next step. Otherwise it will keep sending Status Requests until it receives the correct response.

5. The folder being deleted has its state changed from "delete pending" to "delete now" and the folder is deleted.

6. Another hierarchy replication message is sent out, indicating that the folder has been fully deleted.

> ***Further Information***
>
> This means that in ESM, you do not have to wait for a folder to be fully replicated before removing a replica.
>
> PF replication handles the case where you "move" the replica of a folder by adding a new store, and removing the old store at the same time.

Replication State Tables

Every replicated folder (including the hierarchy) has a Replication State Table, which holds details about the last CN sent by the local copy of a folder (plus additional information required to send a replication message), and the CNSets of all the other replicas of that folder

Every time a replication message is sent out, the CNSets from the Replication State Table *for all the replicas of the folder* are included with the message.

The replication State Tables themselves do not replicate, but the data about all the CNSets for that folder do.

This is how Public Folder Stores learn about what data other replicas of a folder hold.

Replication ID

Each server tracks updates from other server using a Replication ID (ReplID). ReplIDs are calculated locally; therefore stores do not have the same ReplIDs across multiple servers.

Example

This is PFREP57's Replication State Table's replication IDs for a folder

```
REPLID: 10, GUID 4758C926-5CA5-4BA4-AC3A-58A536E9FB58 (/o=Lake
District/ou=Grasmere/cn=Configuration/cn=Servers/cn=PFREP57/cn=Micr
osoft Public MDB)
REPLID: 9, GUID B8E20FA8-9826-43E2-A969-C0D7C7B9B964 (/O=Lake
District/OU=Windermere/cn=Configuration/cn=Servers/cn=PFREP56/cn=Mi
crosoft Public MDB)
REPLID: 5, GUID C19777D7-743B-49E9-A6F5-927AB6E80966 (/o=Lake
District/ou=Windermere/cn=Configuration/cn=Servers/cn=PFREP55/cn=Mi
crosoft Public MDB)
```

PFREP56's might look like this.

```
REPLID: 3, GUID 4758C926-5CA5-4BA4-AC3A-58A536E9FB58 (/o=Lake
District/ou=Grasmere/cn=Configuration/cn=Servers/cn=PFREP57/cn=Micr
osoft Public MDB)
REPLID: 6, GUID B8E20FA8-9826-43E2-A969-C0D7C7B9B964 (/O=Lake
District/OU=Windermere/cn=Configuration/cn=Servers/cn=PFREP56/cn=Mi
crosoft Public MDB)
REPLID: 10, GUID C19777D7-743B-49E9-A6F5-927AB6E80966 (/o=Lake
District/ou=Windermere/cn=Configuration/cn=Servers/cn=PFREP55/cn=Mi
crosoft Public MDB)
```

The ReplIDs are calculated locally.

Example of Replication State Tables & CNSets

In this example, there are 3 replicas of a folder on Servers A, B & C. For simplification, the ReplIDs of the servers are A, B &C and are the same for each server (i.e. Server A sees ReplIDs of A, B & C for Servers A, B & C respectively, and so do Servers B & C).

Simple Replication

Initially the folders are in sync:

ReplID	CNSet details stored on Server A	CNSet details stored on Server B	CNSet details store on Server C
A	Last CN sent A-100	CNSet A-1, A-100 CNSet B-1, B-50 CNSet C-1, C-10	CNSet A-1, A-100 CNSet B-1, B-50 CNSet C-1, C-10
B	CNSet A-1, A-100 CNSet B-1, B-50 CNSet C-1, C-10	Last CN sent B-50	CNSet A-1, A-100 CNSet B-1, B-50 CNSet C-1, C-10
C	CNSet A-1, A-100 CNSet B-1, B-50 CNSet C-1, C-10	CNSet A-1, A-100 CNSet B-1, B-50 CNSet C-1, C-10	Last CN sent C-10

A change is made to the folder on Server A. The replication message is successfully delivered to Servers B & C. The State Tables will now look like this:

ReplID	CNSet details stored on Server A	CNSet details stored on Server B	CNSet details store on Server C
A	**Last CN sent A-101**	**CNSet A-1, A-101** CNSet B-1, B-50 CNSet C-1, C-10	**CNSet A-1, A-101** CNSet B-1, B-50 CNSet C-1, C-10
B	CNSet A-1, A-100 CNSet B-1, B-50 CNSet C-1, C-10	Last CN sent B-50	CNSet A-1, A-100 CNSet B-1, B-50 CNSet C-1, C-10
C	CNSet A-1, A-100 CNSet B-1, B-50 CNSet C-1, C-10	CNSet A-1, A-100 CNSet B-1, B-50 CNSet C-1, C-10	Last CN sent C-10

Another change is now made to the folder on Server B and replicates to A & C.

ReplID	CNSet details stored on Server A	CNSet details stored on Server B	CNSet details store on Server C
A	Last CN sent A-101	CNSet A-1, A-101 CNSet B-1, B-50 CNSet C-1, C-10	CNSet A-1, A-101 CNSet B-1, B-50 CNSet C-1, C-10
B	**CNSet A-1, A-101** **CNSet B-1, B-51** CNSet C-1, C-10	**Last CN sent B-51**	**CNSet A-1, A-101** **CNSet B-1, B-51** CNSet C-1, C-10
C	CNSet A-1, A-100 CNSet B-1, B-50 CNSet C-1, C-10	CNSet A-1, A-100 CNSet B-1, B-50 CNSet C-1, C-10	Last CN sent C-10

As Server C has yet to send its Status (via a replication message), Servers A & B do not know about the changes to the replication State Table on Server C. They will have to wait for either a replication message, or a Status Message from Server C.

CNSets and Backfill

Continuing on, another change is made to Server A, but this time the replication message fails to be delivered to Servers B & C.

ReplID	CNSet details stored on Server A	CNSet details stored on Server B	CNSet details store on Server C
A	**Last CN sent A-102**	CNSet A-1, A-101 CNSet B-1, B-50 CNSet C-1, C-10	CNSet A-1, A-101 CNSet B-1, B-50 CNSet C-1, C-10
B	CNSet A-1, A-101 CNSet B-1, B-51 CNSet C-1, C-10	Last CN sent B-51	CNSet A-1, A-101 CNSet B-1, B-51 CNSet C-1, C-10
C	CNSet A-1, A-100 CNSet B-1, B-50 CNSet C-1, C-10	CNSet A-1, A-100 CNSet B-1, B-50 CNSet C-1, C-10	Last CN sent C-10

However, another change on Server A does replicate through.

ReplID	CNSet details stored on Server A	CNSet details stored on Server B	CNSet details store on Server C
A	**Last CN sent A-103**	**CNSet A-1, A-103** **CNSet B-1, B-51** CNSet C-1, C-10	**CNSet A-1, A-103** **CNSet B-1, B-51** CNSet C-1, C-10
B	CNSet A-1, A-101 CNSet B-1, B-51 CNSet C-1, C-10	Last CN sent B-51 Backfill CNSet A-102, A-102	CNSet A-1, A-101 CNSet B-1, B-51 CNSet C-1, C-10
C	CNSet A-1, A-100 CNSet B-1, B-50 CNSet C-1, C-10	CNSet A-1, A-100 CNSet B-1, B-50 CNSet C-1, C-10	Last CN sent C-10 Backfill CNSet A-102, A-102

Both Server B & C realize a CNSet is missing and write entries into the backfill array.

When the backfill timeout occurs Server B & Server C will send the requests to Server A, because Server A is the only server that has the missing CNSet.

For more information on backfill with multiple servers see **Considerations for Larger Topologies**

Considerations for Larger Topologies

The previous section looked closely at individual replication actions. This next part will look at how the replication process scales when replicas of folders exist on many servers.

Sending Replication Messages to Multiple Stores

The store relies on email transport for delivery of messages. It makes no attempt to split replication messages based on topology details. If the content of a folder is modified and it has 5 other replicas, then a single replication message will be generated and addressed to all 5 other stores.

It is up to transport to determine how to route and deliver and the message.

However, the replication engine will look at the versions of the Stores on the replica list. It uses this information to determine whether certain permissions properties need to be replicated or not. For more information see **Folder Permissions**.

Choosing a Server to Backfill from

When a store determines that is missing a CNSet, either through normal replication or Status Messages, it needs to send a backfill request.

When the backfill timeout expires the store compares the CNSets it is missing for a folder with the knowledge it has about the CNSets that other stores have, from the folder's Replication State Table.

If the Replication State Table indicates that another store has all the missing CNSets then a backfill request will be sent to this store. In the case of multiple stores having the missing data then the lowest cost store is chosen, and Exchange 2000 is chosen over Exchange 5.5.

If the Replication State Table shows that no single store has all the missing CNSets then the store will send multiple backfill requests to multiple stores until all the entries in the backfill array for that folder have been received.

If there is no response to a backfill request within the allowed time interval, then the store the request was sent to will be marked "inactive" and the request sent to a different store.

If possible Exchange 2000 will backfill from another Exchange 2000 server, before attempting to backfill from an Exchange 5.5 server.

Status Requests to more than one server

Status requests are broadcast to all servers in the replica list.

However, only 2 local site stores and 2 off site stores will be placed in responder list (inter site store choice is based on cost).

These stores will respond to the Status requests and send any missing CNSets back to the requestor.

Complications and problems

These are a few examples of when backfilling could take a long time. Normally this can occur when new servers join the organization and transport links have yet to be set up to the rest of the organization.

Backfilling from out of date Server

If a backfill request is sent to a server that does not have the missing data (for example the server the backfill request was sent to had recently had an old backup restored), then the backfill request will not be satisfied.

The store will have to send multiple backfill requests that can take many hours or even days.

Sending Status Requests to a new server

If the server we send the initial Status request to is itself a new store then it may only have Folder 1-1. In which case the stores will appear in sync with each other, even though they are out of sync with the rest of the organization.

This problem will eventually get resolved as updates arrive from other stores in the organization, but because the initial request was satisfied, the subsequent backfills will take hours or even days.

No transport link is available

This will most often occur when a new routing group is created. Install will start the Exchange 2000 Public Store before there is any transport link to the rest of the organization.

The store will send out its Status Request on startup, but will get no response. It will then fall back to using the retry schedule before sending further Status Requests.

Once the transport link is established the server will either successfully send a Status Request, or updates and Status messages from other stores will indicate that it needs to backfill. As the initial Status Request was lost, the data backfill could take hours or even days.

RUS has not stamped mail attributes on Store

If is possible for a store to attempt to send a Status Request, before its directory object has been stamped with the mail attributes. This will result in the replication message NDR'ing.

Again, because the initial Status Request failed, the store may take hours or even days to get back in sync.

In all these cases either the initial replication messages were lost, or the store requested information from another store, which in turn had no information about public folders.

Eventually these situations will resolve themselves as other servers indicate they are missing data.

Default Replication Event Times

This table shows some of the more common timers associated with replication events. The main replication task thread will spawn additional worker threads to handle replication tasks when these times are reached. If there is nothing to replicate then the thread simply exits (i.e. if there are no hierarchy changes to replicate, then no replication message is generated!)

Replication Event	Default Timeout	Comments
"Replication Expiry"	24 hours	How often folders are checked for expiry.
"Replication Send Always"	15 min	This is the default “Replicate Always” value, and is how often the store checks to see whether it needs to replicate content. Can be adjusted through ESM
"Replication Send Folder Tree"	5 min	This is how often the store checks to see whether a hierarchy replication message needs to be sent.
"Replication Send Status Timeout"	24 hours	This is how often the store checks to see if a Status message for a folder should be sent.
"Replication Timeout"	5 min	How often the store will check to see if any backfill timeouts have expired.
"Replication New Replica Backfill Request Delay"	15 min	The time delay used before sending a backfill request for a new folder replica when the data is available in the same site
"Replication Short Backfill Request Delay"	6 hours	The time delay used before sending a backfill request when the data is available in the same site
"Replication Long Backfill Request Delay"	12 hours	The time delay used before sending a backfill request when the data is not available in the same site
"Replication Short Backfill Request Timeout"	12 hours	The timeout value used when retrying sending a backfill request when the data is available in the same site.
"Replication Long Backfill Request Timeout"	24 hours	The timeout value used when retrying sending a backfill request when the data is not available in the same site
"Replication Short Backfill Request Timeout Retry"	24 hours	The timeout value used when sending a backfill request when the data is available in the same site and this is a retry of a previous backfill request
"Replication Long Backfill Request Timeout Retry"	48 hours	The timeout value used when sending a backfill request when the data is not available in the same site and this is a retry of a previous backfill request

Default Replication Values

This table shows some of the other default values used in Public Folder Replication.

Description	Value	Comments
"Replication Folder Count Limit"	20	Max number of folders to pack in a hierarchy replication message
"Replication Deleted Folder Count Limit"	500	Max number of folder deletes to pack in a hierarchy replication message
"Replication Message Count Limit"	100	Max number of messages to pack in a content replication message
"Replication Message Size Limit"	300 Kb	Max replication message size Can be set through admin See **Transport and Routing** for more information

Folder Permissions

This section describes the changes in Folder Permissions between Exchange 5.5 and Exchange 2000. This is especially important during migration and co-existence between Exchange 5.5 and Exchange 2000.

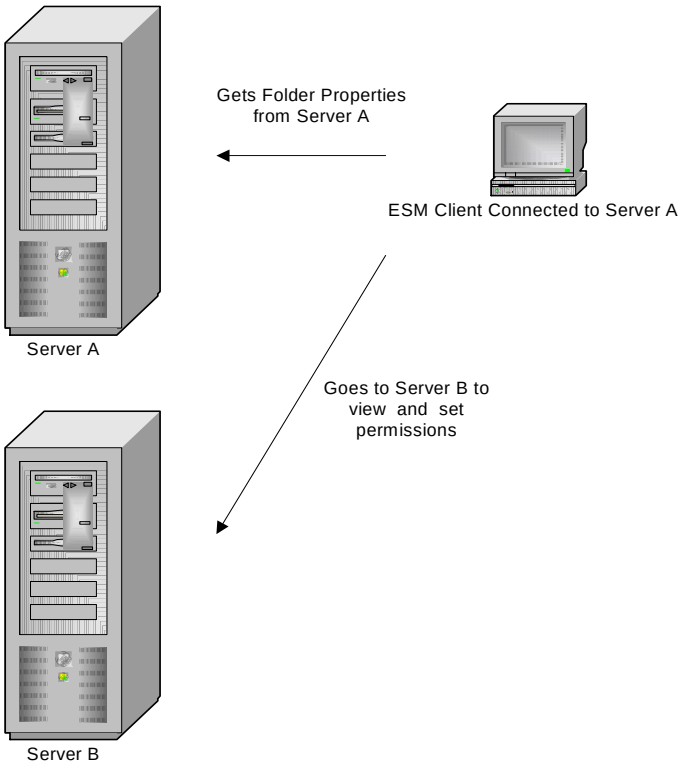
It covers details of the new property tags and differences between MAPI TLH and App TLH trees, how distribution lists are handled and item level security.

For specific details of interoperability between Exchange 2000 and Exchange 5.5 permissions see the section **Replication Co-existence with Exchange 5.5**

As perissions are a folder property, they are replicated as part of the hierarchy. Even "ghosted" folders will have an **Access Control List** (ACL).

Note

When you view a folder's permissions (either through ESM or a client) you will actually connect to a content replica of the folder, not a ghosted folder. In the example below, the administrator is altering permissions on a folder via ESM, which is connected to Server A's public folder tree. When the administrator goes to view and set the permissions, ESM connects to a Store that has an actual replica of the folder.

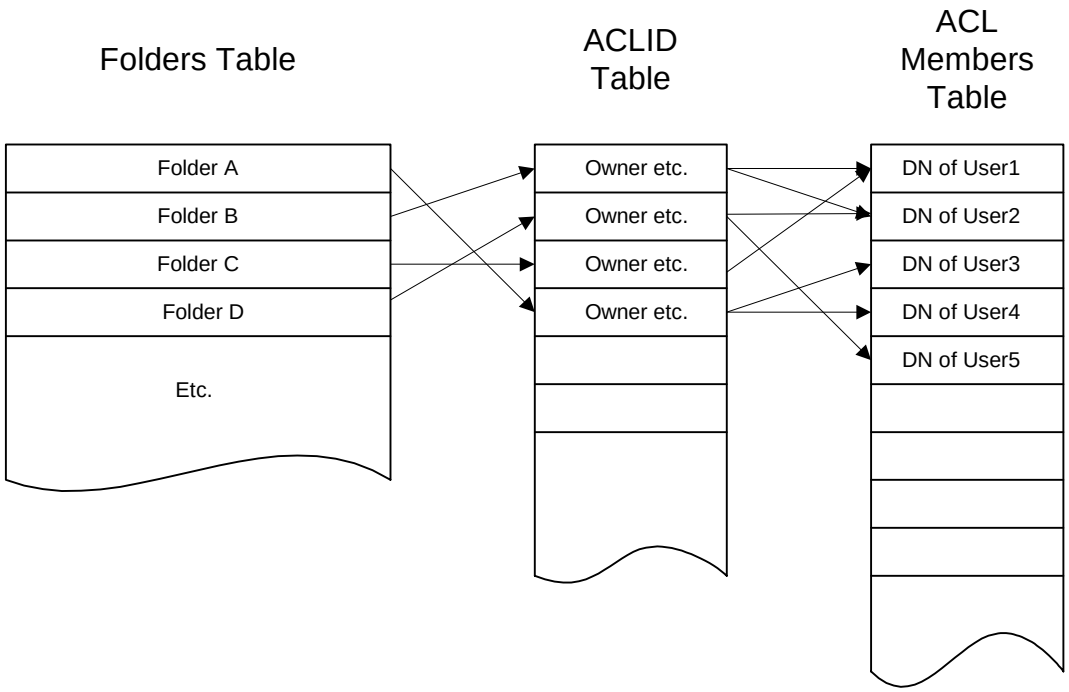


ACL Storage

ACLs in Exchange 5.5

In Exchange 5.5 ACLs on folders were stored in an ACLID table. This table in turn pointed to an ACL Member table, which held the DNs of the objects ACL'd on the folder.

This helped to preserve single instance storage of ACL members in the Store.



There was no ACL property, as such, on the folder itself. Instead the access rights were checked depending on ACLID of the folder.

However, in order to replicate permissions a property is used which holds a binary "blob" of the ACL data: **ptagACLData** (there is also a ptagExtendedACLData for additional permissions: moderators etc.).

The receiving store would then update the ACL tables based on the data replicated in from ptagACLData.

ACLs in Exchange 2000

Permissions have actually been simplified in Exchange 2000.

ACLs in Exchange 2000 are now stored directly as a property of the folder: **ptagNTSD**, which holds the NT Security Descriptors of the users or groups that can access the folder.

Mailboxes in Exchange 2000 are no longer separate from the user. Therefore instead of allowing access to a folder depending on the DN of the mailbox, Exchange 2000 controls access to folder based on the NT Security ID of the logged on user – similar to the way NTFS handles its access control.

This is very important. Most of the problems with users unable to access folders, or even see them, will be caused by problems here. This can be especially dangerous in mixed Exchange 5.5 and Exchange 2000 environments

New ACL ptags

Exchange 2000 has introduced two additional ptags for storing security information:

ptagNTSD (PR_NT_SECURITY_DESCRIPTOR)

ptagAdminNTSD (PR_ADMIN_SECURITY_DESCRIPTOR)

ptagNTSD

This is the "new", richer ACL set that is used by Exchange 2000. The permissions map closely to the NTFS permission set. All folders on Exchange 2000 will have this property

ptagAdminNTSD

These are the Administrator privileges on a folder. By default they are not set on individual folders, but are inherited from the root. However, if you set specific Administrator privileges on a folder then this property will be added and replicated with that folder.

Viewing ACLs in Exchange System Manager

If the folder is a MAPI folder, the MAPI – like permissions will be displayed when viewing the client permissions.

To view the "raw" NTSD permissions on a MAPI folder hold "CTRL" when clicking on permissions in ESM.

Obviously non-MAPI folders will always show the raw NTSD permissions

Support Issue

Do not use Explorer or ESM CTRL → Client, or Windows Explorer, to set MAPI folder permissions. You will lose the ability to modify the permissions via MAPI clients and ESM. For more information see **Replication Problems**.

Distribution Lists & Security Groups

In Exchange 5.5, distribution lists (DLs) could be used to ACL a folder. In Exchange 2000 & Windows 2000 Active Directory DLs have become Groups

There are two types of Windows 2000 Groups

- Security Groups
- Distribution Groups

Security groups are listed in discretionary access control lists (DACLs) that define permissions on resources and objects. Security groups can also be used as an e-mail entity. Sending an e-mail message to the group sends the message to all the members of the group.

Distribution groups are not security-enabled. They cannot be listed in DACLs. Distribution groups can be used only with e-mail applications (such as Exchange), to send e-mail to collections of users.

The equivalent of an Exchange 5.5 DL is a Universal Distribution Group (UDG). If you use the ADC to replicate Exchange 5.5 DLs to the Active Directory they will become UDGs.

This is fine most of the time. However, UDGs cannot by used to ACL public folders. A Security Group must be used to ACL a Public Folder. Under most circumstances the Store will automatically upgrade a UDG to a Universal Security Group (USG), providing the UDG is in a Windows 2000 Native mode domain.

Converting UDGs to USGs

The Store will automatically attempt to upgrade a UDG to a USG if a UDG is ACL'd on a Public Folder. The converter will enumerate the membership of a UDG and also convert the nested member UDGs.

Important

The UDG **must be in a Windows 2000 native mode domain** to allow the Store to upgrade it to a USG. In a mixed Exchange 2000 / Exchange 5.5 environment, the ADC will display a warning if you are replicating Exchange 5.5 DLs to a non-native mode domain. For more information see **Replication Co-existence with Exchange 5.5**

- Assuming that the UDG is in a Windows 2000 Native mode domain, the Store will upgrade a UDG to a USG in the following circumstances:
- When a UDG is added to the ACL list of a folder (either through a client or ESM)
- When an Exchange 5.5 folder is replicated to Exchange 2000.
- When a previous attempt to upgrade a UDG failed (for example, the UDG was in a Windows 2000 Mixed mode domain, the next time the folder is accessed it will attempt to upgrade it again.

Tip

The conversion function is called in the following scenarios:

- On client access, if the UDG has not been successfully converted before it will attempt to convert it.
- On replication, if a replication message contains an update to the ACL, the UDG conversion function will be called.
- Whenever the ACL on a folder is modified directly (e.g. by ESM or client).

It will not be called repeatedly if the UDG was successfully upgraded. This means that if you ACL a UDG on a folder, let it upgrade to a USG, then set the group **back** to a UDG – it will not upgrade again automatically on client access. It will, however, upgrade if you alter the permission associated with the UDG.

When UDG to USG conversion will fail

UDG to USG conversion will not occur under the following circumstances

- The Windows 2000 domain containing the UDG is in Mixed Mode.
- A previously converted UDG is reset back to a UDG (see above).
- The membership of a UDG has not been replicated.
- Nested UDGs will not be converted if their parent is already a USG*

***Note**

The converter determines whether or not to continue enumerating the members based on the type of Group. If a Group is a USG, it will not attempt to enumerate any nested groups to see if they also require converting. This is how the converter determines whether or not to continue walking down the membership list. Otherwise every time an ACL changed on a folder we would keep enumerating all the membership of a group – which would severely impact both Store performance and Windows 2000 GCs.

Therefore the Store will also not convert a UDG to a USG if:

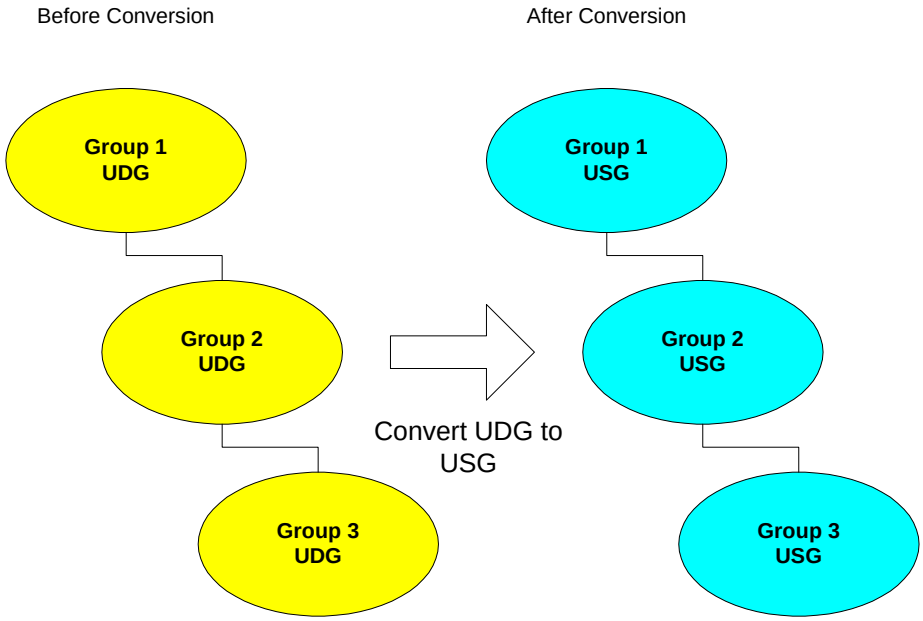
- The administrator manually converts a parent UDG without converting the nested members.
- A UDG is added to the membership list of a USG (either before or after the USG was ACL'd on folder).

Example

Group 3 is a member of Group 2, which is a member of Group 1

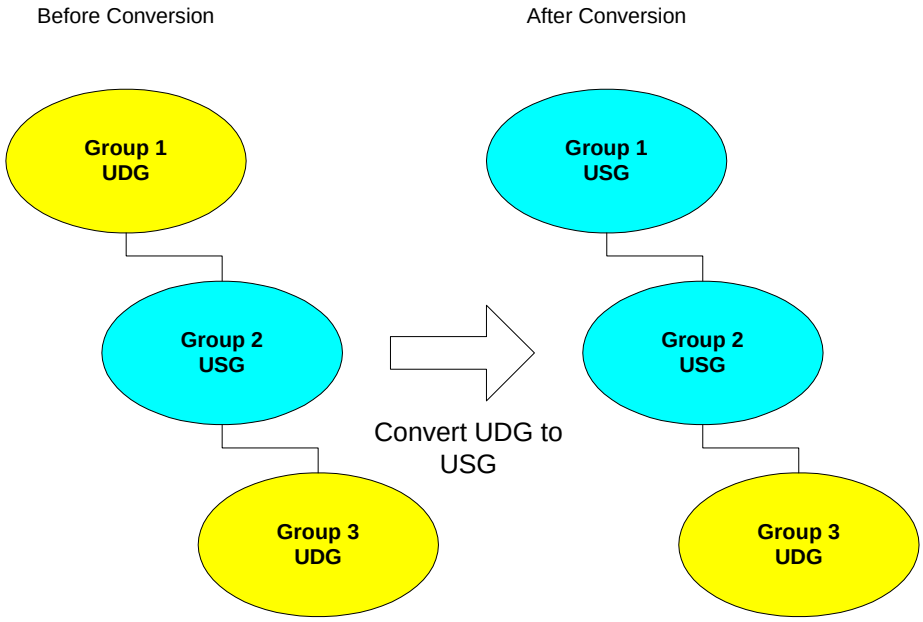
Case 1

Group 1 is placed on the ACL list of a folder. The converted will convert Group 1 to a USG and enumerate the members and convert Groups 2 & 3



Case 2

The administrator has already changed Group 2 to a USG. Group 1 is then placed on the ACL list of a folder. The converter will only enumerate as far as Group 2. So Group 3 will never be converted to a USG, and its members will not be able to access the folder.

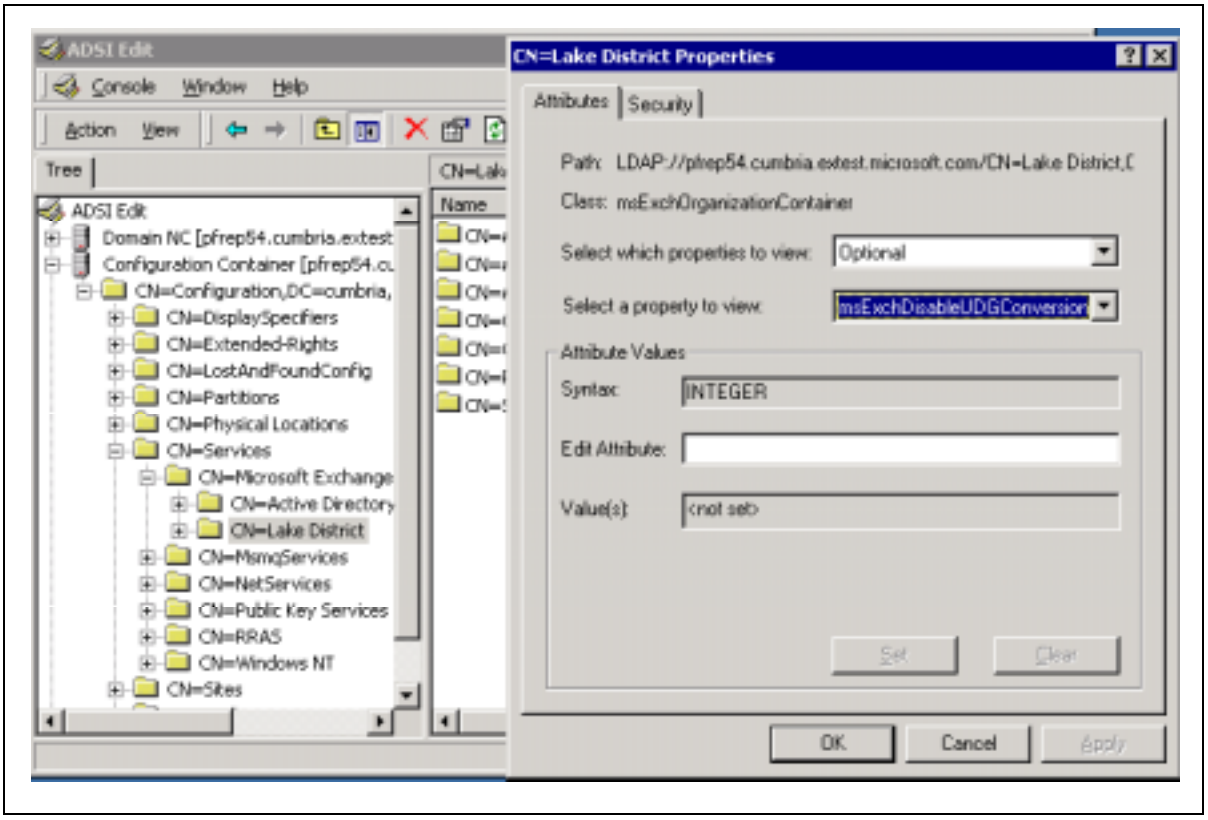


Preventing UDG to USG conversion

In some scenarios customers may wish to avoid token bloat, especially when doing initial replication and migration from Exchange 5.5

You can modify the directory entry msExchDisableUDGConversion (at the Org level) to alter the UDG to USG conversion logic.

```
msExchDisableUDGConversion = 0,1 or 2
```



It can take 3 values:

- 0 or not present (default) – Convert UDG to USG using standard behavior.
- 1 – Convert only if not a client (not really a client, but everything other than replication or upgrade).
- 2 – Never convert (only if you have no intention of retaining old DLs, and want to do everything manually).

Tip

Unless you have a really strong reason to prevent UDG to USG conversion, leave this alone.

Replicating Permissions

Permissions replicate as part of the Hierarchy (they are properties of a folder after all). Depending on the types of permissions and where they are replicating to or from will have an effect on which permissions are replicated.

This topic will also be covered in the next section on Exchange 2000 and Exchange 5.5 co-existence, see **Replication Co-existence with Exchange 5.5**

Replication between Exchange 2000 servers only

If the replication message is not being addressed to any Exchange 5.5 servers (i.e. there are no Exchange 5.5 servers in the replica list), the store will not replicate ptagACLData or ptagExtendedACLData – there is no need.

Therefore, an App TLH will never replicate ptagACLData – in fact it will never use it at all.

Replication between Exchange 2000 and Exchange 5.5 servers

From Exchange 2000 to Exchange 5.5

If a replication message sent from an Exchange 2000 server has Exchange 5.5 Public Stores on its replica list, ptagACLData & ptagExtendedACLData will be calculated and included in the replication update (as an Exchange 5.5 server does not understand ptagNTSD).

Exchange 2000 will always send ptagNTSD and ptagAdminNTSD regardless of whether a receiving server is Exchange 5.5. This means that these property tags will exist on folders on Exchange 5.5, but as Exchange 5.5 doesn't know anything about these properties it will simply ignore them.

From Exchange 5.5 to Exchange 2000

When Exchange 2000 receives ptagACLData it must convert this to NT SIDs and update ptagNTSD accordingly. This is done using a set of temporary tables similar to the ACLID tables in Exchange 5.5. **Once this has completed successfully the temporary tables are removed.**

If it fails to upgrade (for example, due to a user in ptagACLData not existing in the Active Directory), then only **Owners** will be promoted into ptagNTSD. The ptagACLData details will remain (in the temporary ACLID table rows) and the store will attempt to upgrade it next time the folder is accessed. Anonymous permissions will not be set either.

The failure to upgrade the ACL from ptagACLData to ptagNTSD is the most common reason for permissions problems in mixed mode Exchange Organizations.

For more information see **Replication Co-existence with Exchange 5.5**.

If the replication message originated from an Exchange 5.5 server, the receiving replication engine on Exchange 2000 will drop the following properties:

- ptagNTSD
- ptagAdminNTSD

This is to prevent a security hole. The security on these properties is not enforced on an Exchange 5.5 server. Therefore, someone who doesn't have permissions to manipulate them on an Exchange 2000 server could manipulate them on an Exchange 5.5 server. This has no effect on Exchange 5.5, but could be a problem if they ever replicated back to Exchange 2000. Essentially, it might be possible for a malicious user to add themselves to the folder Administrator privileges (ptagAdminNTSD) via Exchange 5.5. By disallowing these properties to be replicated in from an Exchange 5.5 server, any changes to these properties made on Exchange 5.5 will be dropped, so there is no security hole.

The only problem with this is that if an Exchange 2000 server were to backfill a folder or hierarchy from an Exchange 5.5 server, changes to these properties would be lost (this will not create a security hole, but might remove someone from the permissions list. In this event the users would have to be re-ACL'd on the folder).

Summary of Permissions Properties

Property	Description	Exchange 2000 replicating to only Exchange 2000 Stores	Exchange 2000 replicating when Exchange 5.5 stores exist on the replica list.	Exchange 5.5 replicating out
ptagNTSD	The ACL for Exchange 2000 folders. This controls all access to folders on Exchange 2000.	Is replicated.	Is replicated. Therefore, Exchange 5.5 servers will get this property set on their copies of the folder. To Exchange 5.5 this is just another property and is ignored.	Is replicated (if it exists). Although Exchange 5.5 doesn't use this property, if Exchange 5.5 replicates the folder properties out, then this property will be included with all the others. However, Exchange 2000 will not accept this property and drops it.
ptagAdminNTSD	The ACL for Administrative rights. This controls who can administer folders on Exchange 2000 (i.e. set replica lists, age limits etc.)	Is replicated	Is replicated. Therefore, Exchange 5.5 servers will get this property set on their copies of the folder. To Exchange 5.5 this is just another property and is ignored.	Is replicated (if it exists). Although Exchange 5.5 doesn't use this property, if Exchange 5.5 replicates the folder properties out, then this property will be included with all the others. However, Exchange 2000 will not accept this property and drops it.
ptagACLData	A binary blob used by Exchange 5.5 to replicate permissions details (does not actually control access to folders, but is unpacked into ACL tables).	Not used. Will not be replicated.	Is replicated. This property will be calculated (from ptagNTSD) and replicated out; this is so Exchange 5.5 can be made aware of permissions changes.	Is replicated. When Exchange 2000 receives this property, it must be unpacked and the ACLs upgraded into NT SIDs and written into ptagNTSD.

For more information on mixed mode Exchange Organizations see **Replication Co-existence with Exchange 5.5**.

Replication Co-existence with Exchange 5.5

This section describes how Public Folder replication occurs between Exchange 2000 and Exchange 5.5. This is the most complex part of deploying public folders, and can cause the most problems.

The replication engine itself has changed very little between the two versions. However, in order for replication to work, directory details must replicate correctly between the Exchange 5.5 Directory and the Windows 2000 Active Directory, mail must be able to be sent between Exchange 2000 and Exchange 5.5, and users must have permissions replicated correctly.

This section is divided into these parts:

- Directory Replication – ADC Connection Agreements
- Public Folder Replication
- Permissions

ADC Connection Agreements

The Exchange 2000 Active Directory Connector (ADC) is used to replicate objects and containers between the Exchange 5.5 Directory Service (DS) and the Windows 2000 Active Directory (AD).

This will not include extensive details on how ADC replication works.

All three types of ADC connection agreements are used for Public Folder co-existence with Exchange 5.5.

- Configuration CA
- User CA
- Public Folder CA

This section will concentrate on how they affect Public Folder replication

Tip

Always configure User & Public Folder CAs to replicate between W2K DC/GC and an SRS where possible (obviously this can't be done for sites with are pure Exchange 5.5). When an Exchange 2000 server is added to an Exchange 5.5 site, move any existing User & Public Folder CAs to replicate with the SRS.

The SRS uses LDAP Port 379.

This is because SRS' hold the writeable copies of pure Exchange 2000 Admin Groups.

Configuration CA

Config CAs replicate Site / Admin Group configuration objects between Exchange 5.5 and the Active Directory. They are created automatically by setup.

These are some of the important Objects and Attributes that replicate over this CA.

MAPI Stores

Exchange 5.5 ⇔ Active Directory	Comments
Site-MDB-Config (Information Store Site Configuration) ⇔ Admin Group	
Site-Folder-Guid ⇔ siteFolderGUID	Used to identify the site folders for this site
Site-Folder-Server ⇔ siteFolderServer	Name of the Server responsible for hosting the site folders (normally the first server in the Site / Admin Group
Folders-Container ⇔ msExchPfCreation	Location to create public folder's directory entries in Exchange 5.5. If not present then the Recipients container is used. In Exchange 2000 this attribute is read by the store on startup to determine what LegacyExchangeDN must be used by the store when a folder is created in Exchange 2000. This way it will then replicate back to the correct container in 5.5 via the PF CA. For further information see **Exchange 5.5 can use multiple containers for folder directory** entries.
Microsoft Public MDB ⇔ Public Information Store (<server>)	
Obj-Dist-Name ⇔ LegacyExchangeDN	Replicates the Exchange 5.5 Public Store's Obj-Dist-Name to the LegacyExchangeDN of the Store object in the Active Directory
E-mail Addresses ⇔ proxyAddresses	Keeps the Public Folders stores email addresses in sync. Stores replicate by emailing updates to each other. Therefore each store requires an email address.
Home-MTA ⇔ HomeMTA	Replicates the HomeMTA to Exchange 5.5, so the Exchange 5.5 can route replication messages to the Exchange 2000 Store

Application TLH Stores

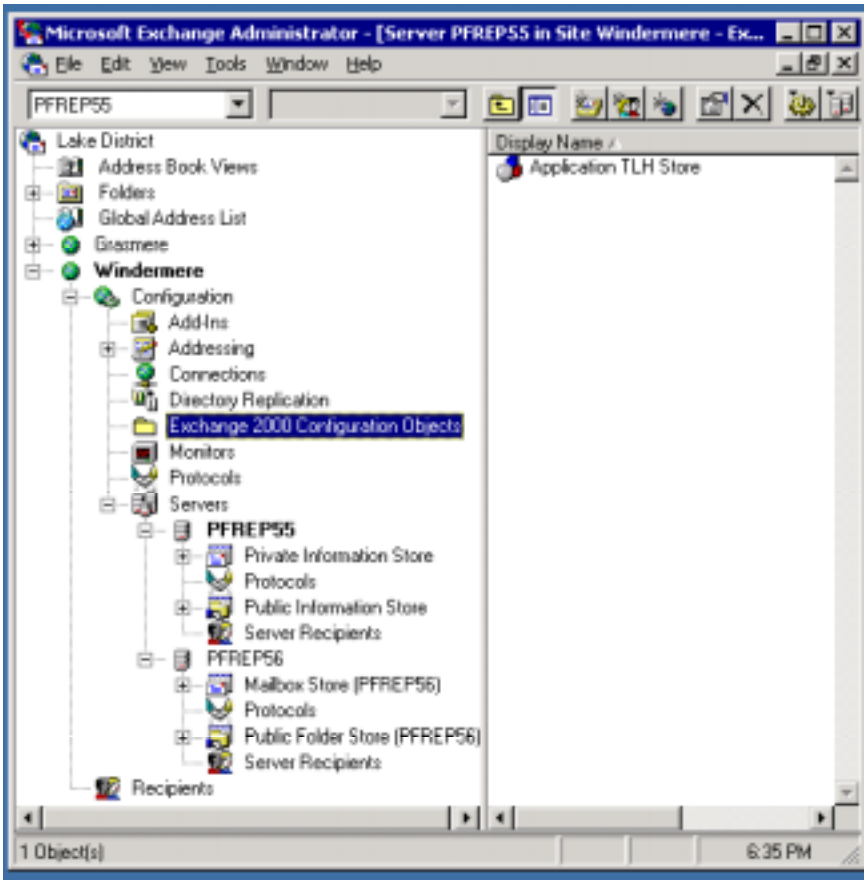
Directory entries for stores belonging to Application TLHs replicate to Exchange 5.5 differently. They must have a directory entry in the Exchange 5.5 DS (to allow 5.5 MTAs to be able to look up their HomeMTA attributes when an Application TLH replication message is being routed via an Exchange 5.5 server.)

However, they can't be placed in the same container as other Exchange 5.5 stores, due to the problems this would cause in Exchange 5.5 (Exchange 5.5 cannot cope with "seeing" multiple Public Stores on a single server).

Instead they are replicated into 5.5 as special custom objects (rather than stores) and placed in their own container "**Exchange 2000 Configuration Objects**". The replication is one-way – the entries never replicate back to the Active Directory.

Example

How an App TLH store looks to Exchange 5.5



The only reason for having these entries is to allow App TLH replication message to be routed via Exchange 5.5 MTAs.

Active Directory → Exchange 5.5	Comments
Application Store Object → Exchange 2000 Configuration Object	
LegacyExchangeDN → Modified Obj-Dist-Name	The LegacyExchangeDN does not map directly to the Obj-Dist-Name (otherwise the App TLH Store object would be in the same container as MAPI Public Stores. Instead the object is placed in the Exchange 2000 Configuration Objects container.
HomeMTA → Home-MTA	This allows the Exchange 5.5 MTA to route replication messages to the correct Exchange 2000 server.
proxyAddresses → Email-Addresses	Replicates the App TLH Store's email addresses to Exchange 5.5
LegacyExchangeDN → X.500 Pilgrim Address	The App TLH Store's LegacyExchangeDN is replicated to an additional X.500 address or "pilgrim" address. E.g. X500:/O=Lake District/OU=Windermere/cn=Configuration/cn=Servers/cn=PFREP56/cn=MICROSOFT PUBLIC MDB92489005

User CA

The User CA is very important in mixed Exchange 5.5 / Exchange 2000 organizations. It is responsible for replicating mailboxes, custom recipients and distribution lists to users, contacts and groups.

This is important for MAPI Public Folders in a Mixed Organization, because Users and DLs can be ACL'd on folders. When the folders replicate to Exchange 2000 the ACLs must be changed to hold SIDs, so the Exchange 2000 store **MUST** be able to find corresponding objects in the Active Directory, or the ACL list will not be set correctly.

- Exchange 5.5 Mailboxes will replicate to Users in the Active Directory
- Exchange 5.5 Mailboxes whose primary NT account is not in the domain will replicate to Disabled Users, with an **Associated External Account** pointing to the original, trusted NT account. The Associated External Account is the **msExchMasterAccountSid** attribute on the W2K AD object

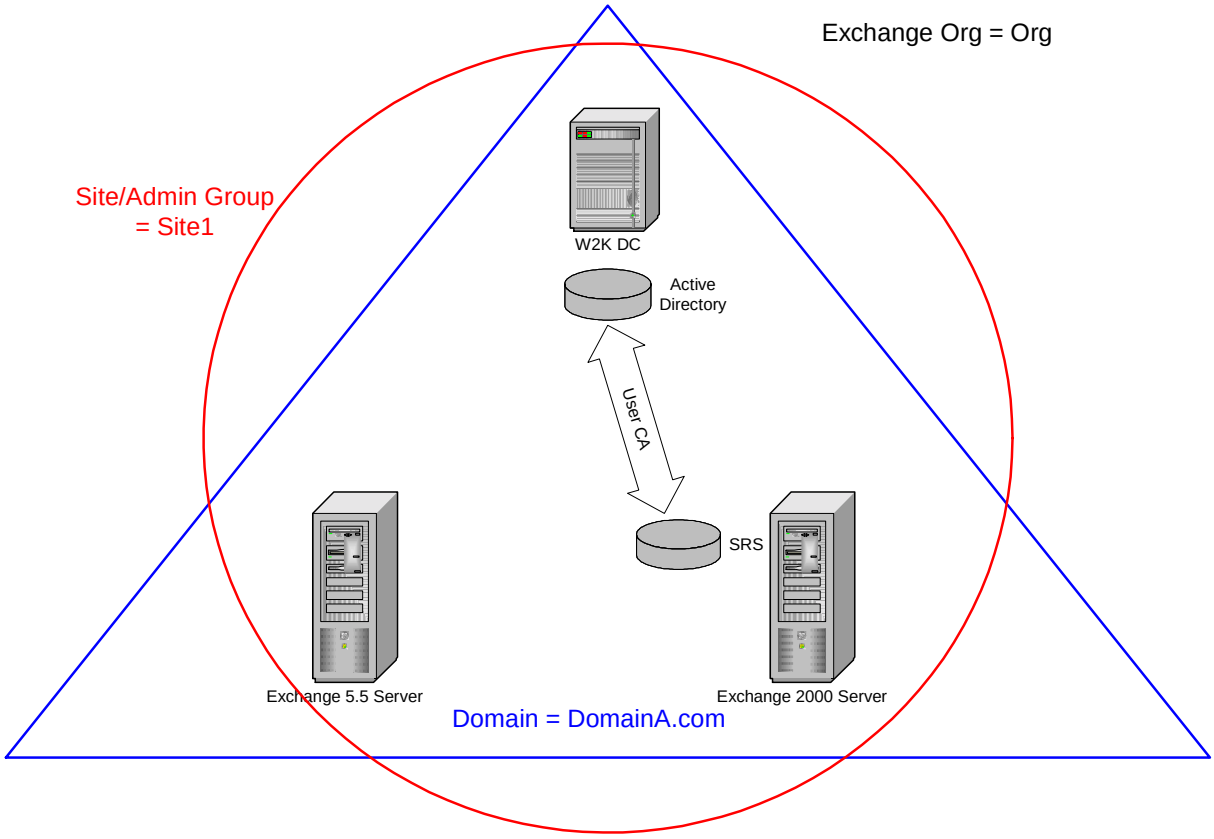
For more information see **Permissions**. It covers how different replication scenarios affect permissions.

The following will list some examples of common User CA scenarios, and the type of W2K Users created. More complex scenarios exist, but the basics will still apply.

Replicating Mailboxes & Users

Scenario 1: Exchange 5.5 Mailboxes' primary NT Accounts are in same Windows 2000 domain as Exchange 2000 Users

This is the simplest scenario of all. No NT4 accounts are involved. The Exchange 5.5 mailboxes will replicate to W2K Users with the correct account information. The W2K accounts are not disabled. Exchange 5.5 DNs ACL'd on folders can be changed to SIDs by the Exchange 2000 Store.



Example

User1 has a mailbox on the Exchange 5.5 server. The mailbox's primary NT account is on the W2K DC. The User CA will replicate the mailbox properties (including the LegacyExchangeDN) to the User in the Active Directory.

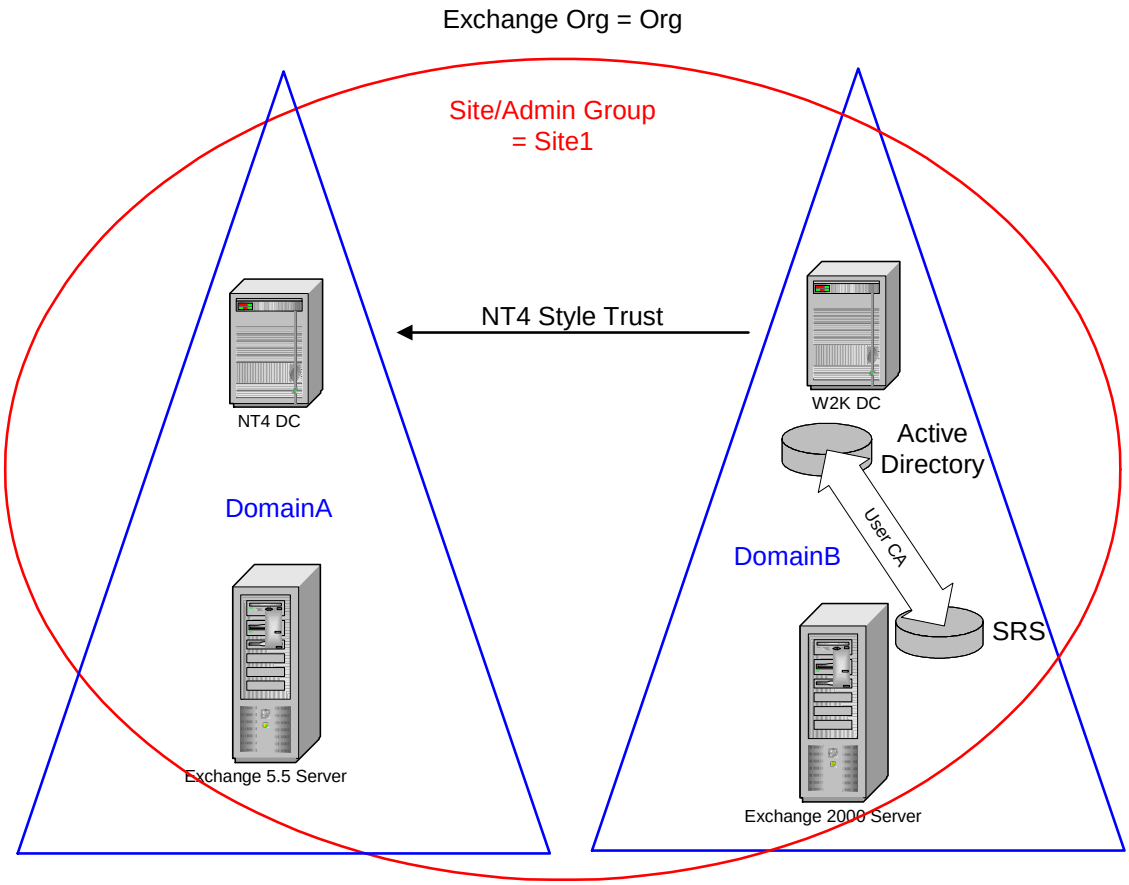
User1 is ACL'd on a Public Folder on Exchange 5.5 so ACL entry will read:

```
"DN:/o=Org/ou=Site1/cn=Recipients/cn=User1 is an Author."
```

When the Exchange 2000 server receives the ACL, it can look up DN:/o=Org/ou=Site1/cn=Recipients/cn=User1 against the LegacyExchangeDNs in the W2K AD and replace the DN stored with the SID of User1's NT account.

Scenario 2: Exchange 5.5 Mailboxes' Primary NT Accounts are in a domain Trusted by the Windows 2000 domain

This is another common scenario. Rather than upgrading an existing NT4 domain to W2K, a separate W2K domain is created, which trusts the NT4 domain. The primary NT accounts of the Exchange 5.5 users are in the NT4 domain.



Example

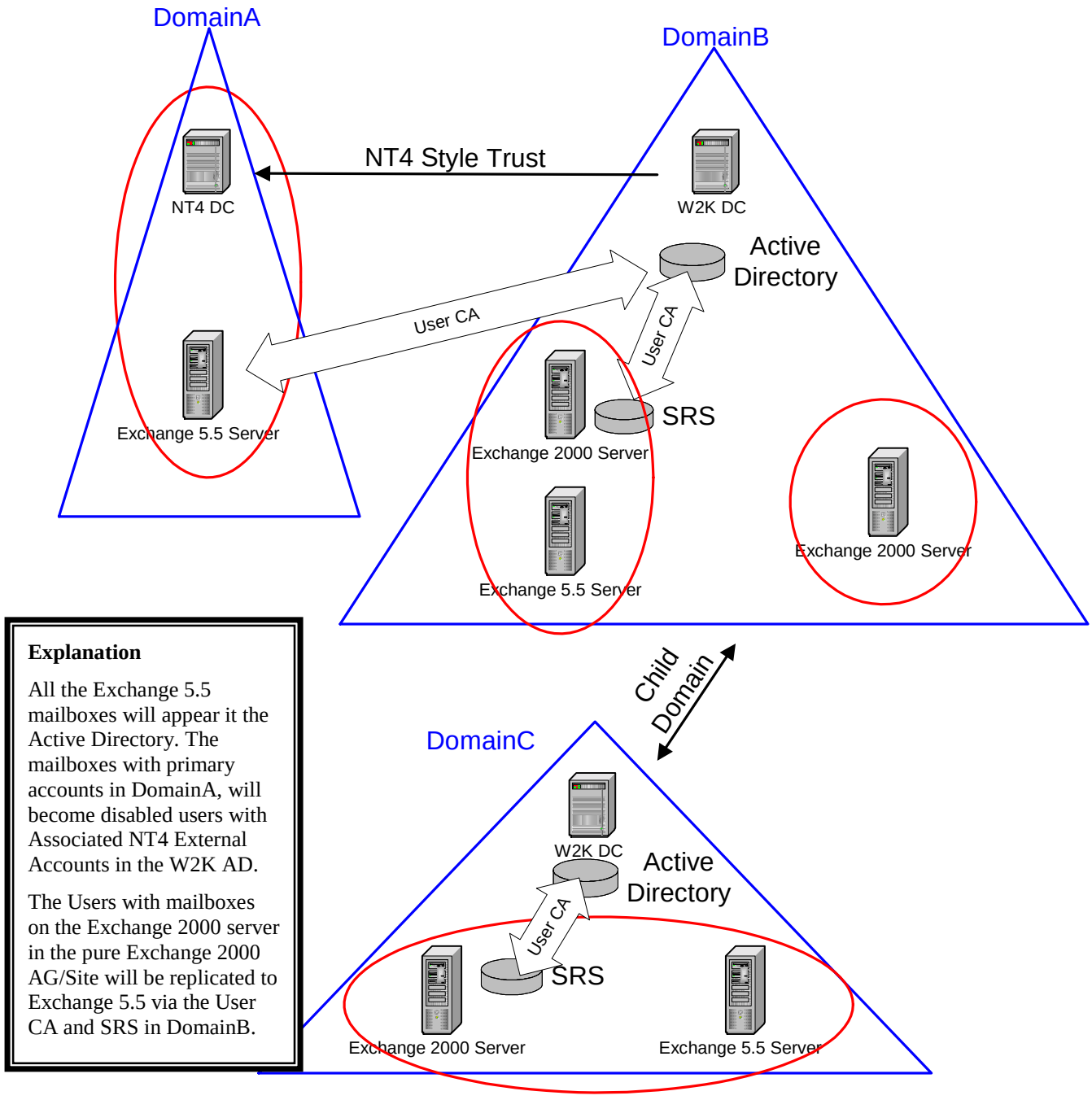
User1 has a mailbox on the Exchange 5.5 server, and their primary NT account is in DomainA.

The User CA will replicate User1's mailbox (including the LegacyExchangeDN) to a **disabled User Account** in the W2K AD (Domain B). This disabled account will have an **Associated External Account** of the NT4 primary account.

User1 is ACL'd on a Public Folder on Exchange 5.5. When this replicates to Exchange 2000, the store will look up the LegacyExchangeDN and find a disabled user. However, because the account has an Associated External Account, the NT4 account will be used on the folder's ACL

Scenario 3. Larger Topologies.

Essentially larger topologies are just extensions of the previous two case.



Explanation

All the Exchange 5.5 mailboxes will appear it the Active Directory. The mailboxes with primary accounts in DomainA, will become disabled users with Associated NT4 External Accounts in the W2K AD.

The Users with mailboxes on the Exchange 2000 server in the pure Exchange 2000 AG/Site will be replicated to Exchange 5.5 via the User CA and SRS in DomainB.

Replicating Distribution Lists

Exchange 5.5 DLs replicate to W2K UDGs, via a User CA

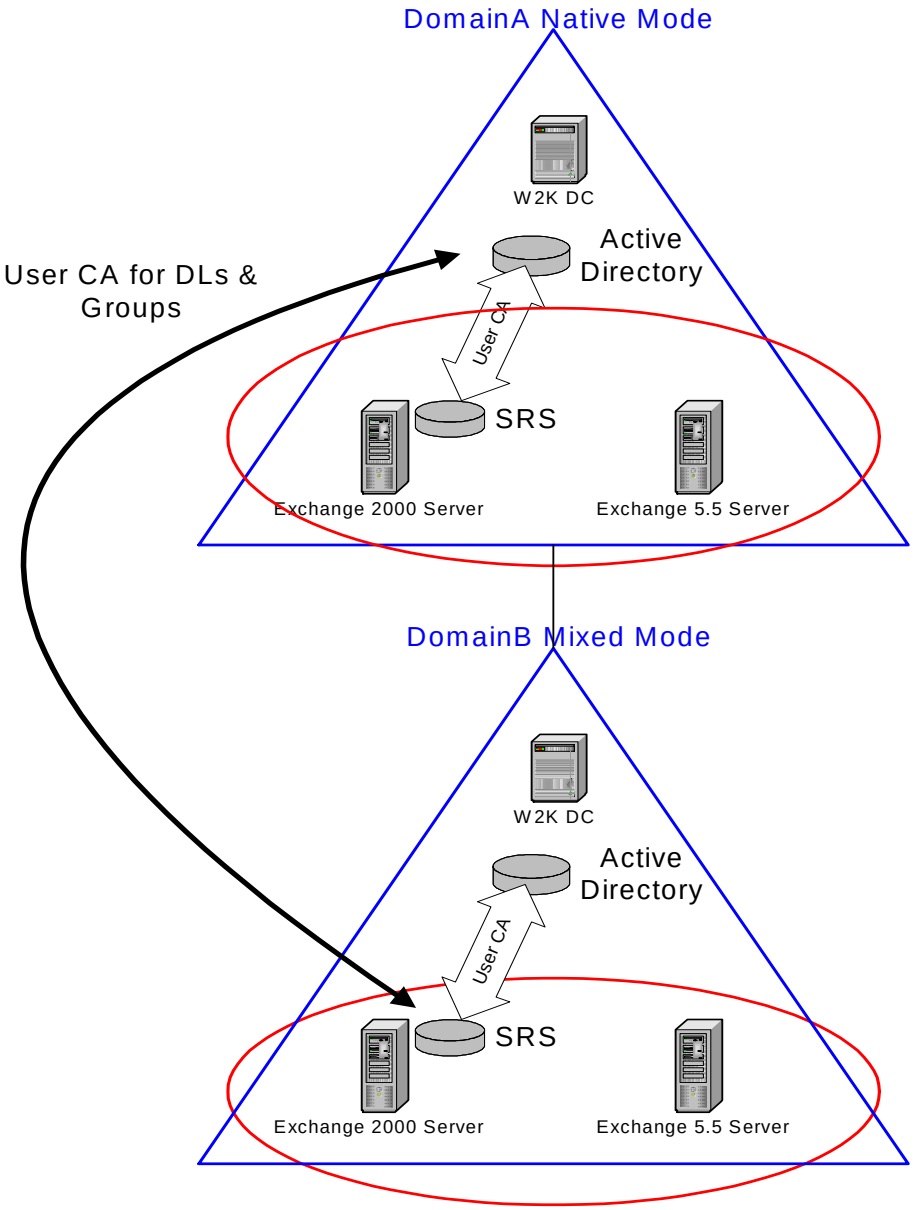
Care should be taken to ensure that User CAs replicate DLs to Native Mode W2K domains. This allows UDGs to be converted to USGs when ACL'd on a folder

Explanation

Domain B is in Mixed Mode, so UDGs cannot be upgraded to USGs if they are ACL'd on a folder.

Instead a separate User CA has been created to replicated just DLs & Groups between the SRS in DomainB and the Active Directory in Domain A.

The other User CA in DomainB has had the option to replicate DLs & Groups unchecked.



Public Folder CA

This is actually the least important of all the CA types for public folder replication. Public Folder connection agreements replicate the actual Public Folders' directory objects between Exchange 5.5 and the AD.

An individual public folder's directory object exists purely so that the public folder can be emailed (this is why in Exchange 2000, mail disabled Public Folders have no directory entry).

> ***Further Information***
>
> Even if there are no plans to email folders, PF CAs should still be set up at the same time as User CAs. This can greatly reduce problems in the future.

However, Public Folder CAs should still be created to all sites in the Organization, for the following reasons:

- Folders created on Exchange 2000 cannot be administered from Exchange 5.5 if they don't have a directory entry in the Exchange 5.5 DS. The Exchange 5.5 admin program expects to find a directory entry for all public folders
- Folders created on Exchange 5.5 will generate errors if administered from Exchange 2000 and they don't have a directory entry in the Windows 2000 AD. The folder has properties stating that it is mail enabled, so ESM will try to find the folders' directory entries. The error can be cleared and the folder still administered (but it gets annoying after a while). Worse an administrator may attempt to re-mail enable the folder and create a separate W2K directory entry. If a PF CA is ever put in place, there will now be two directory entries for the same folder and email to it will NDR.
- Administrators running a DS/IS adjust on Exchange 5.5 can create directory entries incorrectly for Exchange 2000 folders if their directory entries are not replicated. Effectively there would be two separate DS entries (one in the Active Directory, one in the Exchange 5.5 DS) for the same folder. If the directories ever did replicate in the future, public folders could have two directory entries in both the Exchange 5.5 and Windows 2000 Directories. This will prevent emailing to the folder.
- There may be a future requirement for emailing a folder. If all the Exchange 5.5 servers are removed by this time, then there is nowhere to replicate the directory entries from anymore, so the folders will have to be updated manually (or re-mail enabled via a script).

Configuring Public Folder CA

Most of the options for PF CAs are automatically configured. The only options that can be manually set are:

- General - Name of CA, server to run CA. The CA must be two-way
- Connection – Security credentials for connecting to Windows 200 AD and Exchange 5.5 DS
- Schedule

Everything else is calculated automatically

Exchange 5.5 can use multiple containers for folder directory entries

In Exchange 5.5 it was possible to set the directory container that public folder's directory entries were created in. This was the Folders-Container attribute of the Information Store Site Configuration object. By default this attribute was not present, and the Recipients container was used.

The Config CA replicates this property (if set) into the Active Directory (the Admin Group's msExchPfCreation attribute). The store reads msExchPfCreation on **startup** and if present it will use the container specified when creating an Exchange 2000 folder's LegacyExchangeDN. Because the Public Folder CA will create Public Folder directory entries based on their LegacyExchangeDN, the Exchange 2000 folder's directory entries will be placed in the same container as the Exchange 5.5 folder's directory entries.

Replication Via Public Folder CA

The table describes details of how the ADC replicates Public Folders' directory objects via a PF CA.

Replication From Exchange 5.5	Replication From Exchange 2000
Search for Exchange 5.5 Public Folders' DS objects is based from the Site Level. This means that all containers are searched for Public Folders, not just the Recipients container.	Search for Exchange 2000 Public Folders' DS objects in the Microsoft Exchange System Objects container. Public Folders' directory objects are only held in this container in the Windows 2000 AD.
Folder directory objects replicate to the "Microsoft Exchange System Objects" container in the Windows 2000 AD	Folder objects replicate into Exchange 5.5 DS based on their LegacyExchangeDN. The LegacyExchangeDN is set by the store when a folder is created. The stores sets the LegacyExchangeDN attribute based on the value of msExchPfCreation. msExchPfCreation is replicated from the Exchange 5.5 DS attribute Folders-Container. If it is not set then the Recipients container is used.
The Home-MTA and Home-MDB attributes are not replicated (they are meaningless to Exchange 2000)	The HomeMDB and targetAddress attributes are not replicated (they are meaningless to Exchange 5.5)

Troubleshooting Tip

Replication occurs by two stores emailing each other – it is not necessary for the folders to have directory entries for this to work. **Therefore, if there are problems with replication, access permissions or referrals, the PF CA is the last place to look!**

Exchange 5.5 and Exchange 2000 Folder Replication

MAPI Folders

In Exchange 5.5 Public Folders are part of the MAPI TLH (there was only one). Therefore, replication between folders in the MAPI TLH can occur between Exchange 5.5 Public folder stores and Exchange 2000 MAPI Public folder stores.

The Exchange 2000 MAPI Public Stores are replicated into the 5.5 DS and so appear to 5.5 as normal public folder stores – complete with all the email addresses required to email replication messages between them.

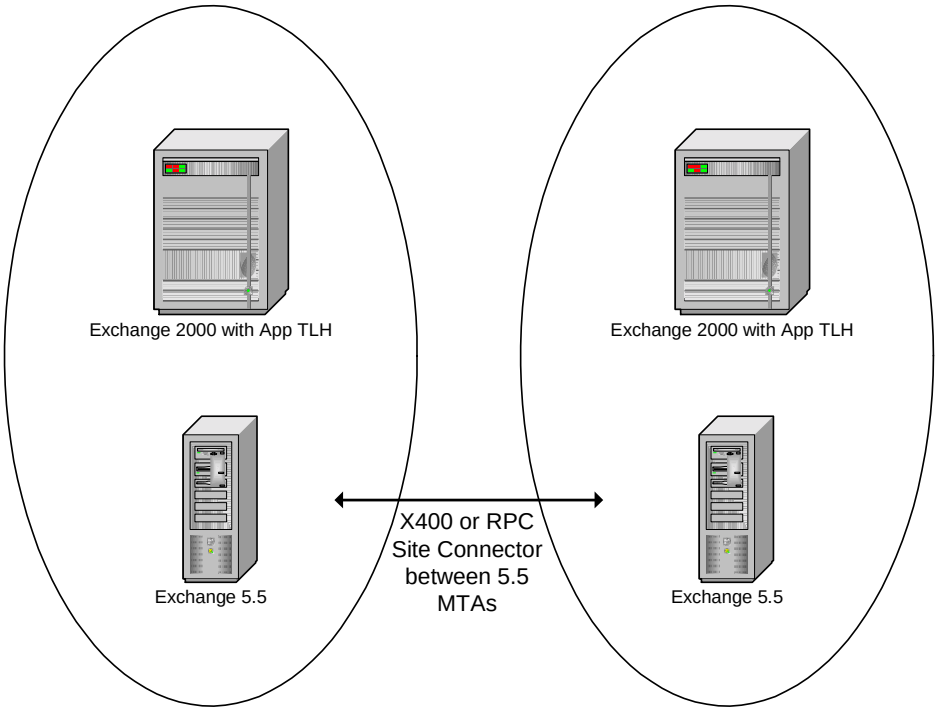
There are no special considerations to take into account in order to replicate MAPI Folders between Exchange 5.5 and Exchange 2000.

Note

This does not mean there can't be any problems. The most common one will be incorrect translation of ACLs from DNs to NT DACLs. As stated previously Exchange 2000 uses NT SIDs for ACL entries. The translation occurs when a folder replicates from Exchange 5.5 to Exchange 2000 (and vice versa). If Exchange 2000 cannot resolve the DN to a W2K user then the permission will be dropped. So although the folder has actually replicated, Exchange 2000 users may not be able to see it due to not having "View" permissions on the folder. See the section **Permissions** for more information.

App TLH folders

Whilst Exchange 2000 cannot replicate App TLH folders to Exchange 5.5 (Exchange 5.5 can only host MAPI TLH stores) it is possible for one Exchange 2000 App TLH store to send a replication message to another App TLH store via an Exchange 5.5 messaging transport (namely the MTA). This is most likely to occur in topologies where member servers are upgraded before bridgeheads, or where an organization uses a predominantly Exchange 5.5 messaging backbone.



The problem occurs when a message destined for an App TLH store is routed into the destination site via an Exchange 5.5 MTA. The address of the message matches the MTA's local Site-Addressing, therefore it knows that the object is local to this site and so performs a lookup in the Exchange 5.5 DS to find the Home-MTA for the recipient.

However, the App TLH stores are not replicated to the Exchange 5.5 DS (due to problems this would cause Exchange 5.5 Public Folder Replication). So the MTA lookup would fail and the replication message NDR'd.

Instead the Config CA replicates the App TLH stores into the 5.5 directories as special objects. These objects are then given an additional X500 proxy address that matches the LegacyExchangeDN name of the Active Directory object (i.e. the X500 address the replication message was emailed to).

Note

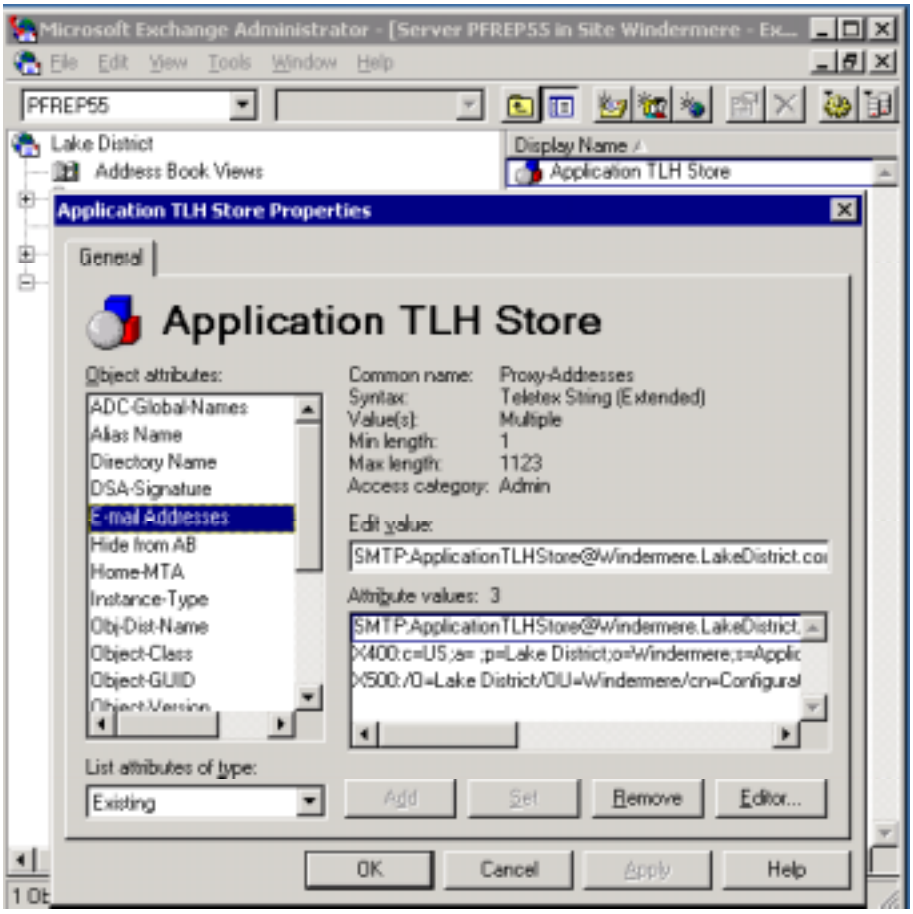
This is exactly the same method used to email "Pilgrim" users in Exchange 5.5

App TLH Store Email-Address attributes

There is an additional X500 proxy address on the object. This is the ExchangeLegacyDN attribute from the Active Directory

Example

These are the Email-Address attributes of an App TLH Store object replicated to Exchange 5.5



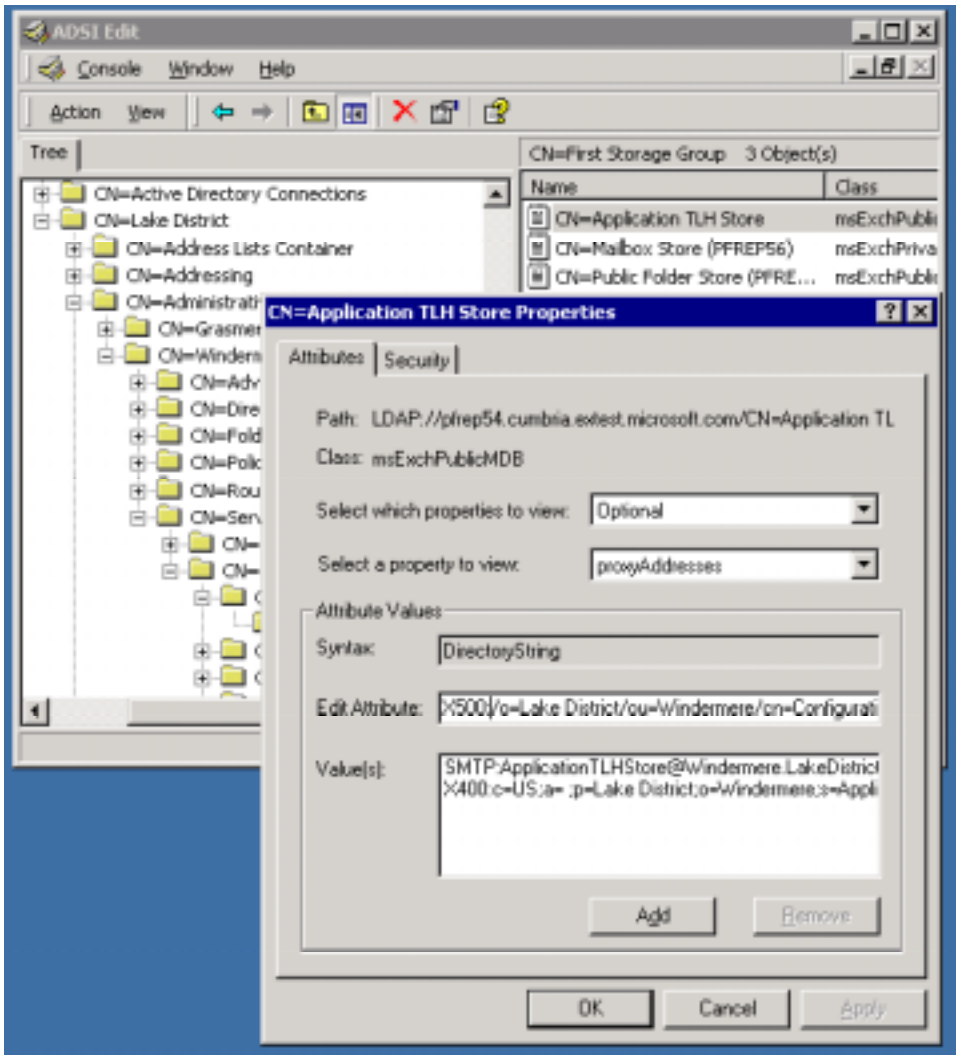
The Exchange 5.5 MTA now has an object to resolve against and read its HomeMTA attribute. The MTA will then replace the Recipient OR address with the Obj-Dist-Name in its directory and route the mail to the Exchange 2000 MTA.

When the message arrives at the Exchange 2000 MTA it will attempt to resolve the DN the message is addressed to. This will fail because the Active Directory has no entry for the Obj-Dist-Name of the entry in the Exchange 5.5 DS. However, it works because the X.400 address has been preserved intact all the way through, and the MTA can find a match on this and deliver the replication message to the correct store*

*Special Instructions for App TLH replication over Exchange 5.5 IMC

The only time the X.400 address is not preserved is if the message has been sent across an Exchange 5.5 IMC. This is a special case and requires the Active Directory entry to be modified. To get this to work the Exchange 5.5 Obj-Dist-Name needs to be added as an additional X.500 proxy to the App TLH store entry in the Active Directory

> ***Notes***
>
> This is only required for replicating App TLHs via an Exchange 5.5 IMC.
>
> The pilgrim address is type X500 of the form:
>
> X500:/obj-dist-name
>
> There is no "." in X500



The full proxyAddresses attribute for the App TLH store now reads:

```
3> proxyAddresses: X500:/o=Lake
District/ou=Windermere/cn=Configuration/cn=Exchange 2000
Objects/cn=Application TLH Store;
SMTP:ApplicationTLHStore@Windermere.LakeDistrict.com; X400:c=US;a=
;p=Lake District;o=Windermere;s=ApplicationTLHStore;;
```

Permissions

> ***Important***
>
> **If you read nothing else, READ THIS SECTION!**

Replicating permissions between Exchange 5.5 and Exchange 2000 is the most hazardous area in mixed mode operation. When everything is set up correctly, they will work. However, as soon as anything goes wrong, it will become immediately apparent – clients will not be able to access or even see folders, owners cannot change permissions and there will be errors in event log.

This section will tie together all the details about permissions covered previously and show how they work and what to look for if there are problems.

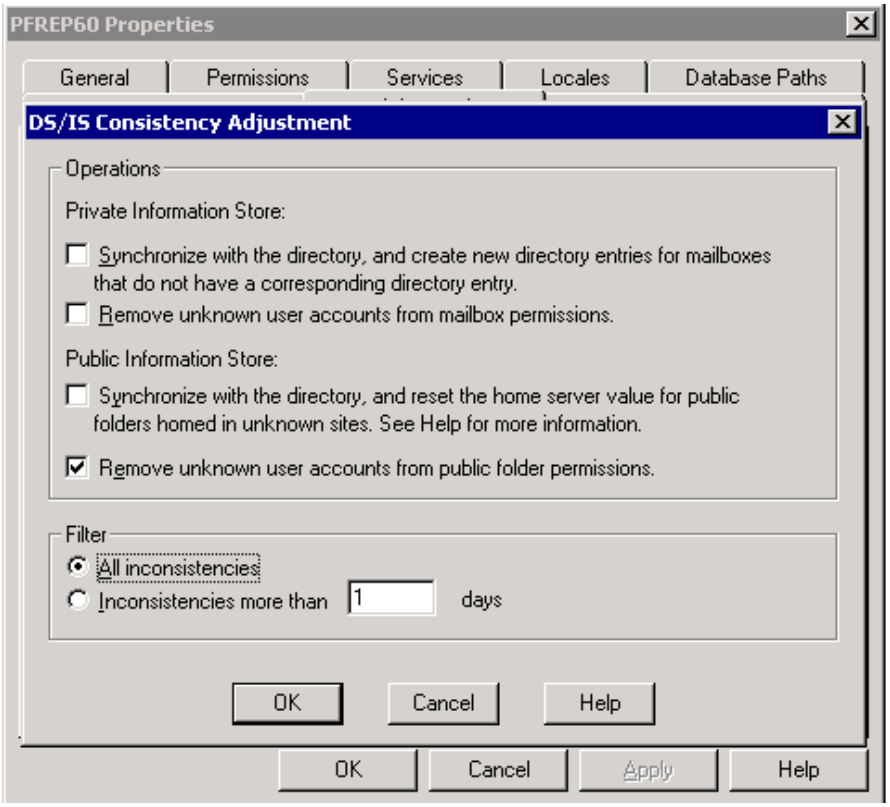
For additional permissions information see the previous section **Folder Permissions**.

Much of this section equally applies to ***upgrading*** a Public Folder store from Exchange 5.5 to Exchange 2000.

DS/IS Adjust

As Exchange 2000 requires all the users ACL'd on folders to have entries in the directory, it is advisable to run DS/IS adjust on Exchange 5.5 before the first Exchange 2000 server is introduced into the Organization. This will remove any deleted Exchange 5.5 users from the permissions lists of folder.

> ***Important***
>
> Only run "Remove unknown user accounts from public folder permissions"
>
> Do not start rehoming folders!



This may need to be done on one server in each Exchange 5.5 site, due to Exchange 5.5's "Limit Administrative Access to Home Site" setting.

Tip

When a mailbox was deleted in Exchange 5.5, any permissions the mailbox had on folders were not removed. Instead the ACL would show an entry of:

```
/o=<org>/ou=<site>/cn=<container>/cn=user
```

instead of the display name.

This had no ill effects on Exchange 5.5. However, this will cause problems for Exchange 2000, because it will be unable to upgrade this DN to an NT SID. Running DS/IS adjust removes these unknown users accounts from Exchange 5.5 folders.

Replicating Permissions From Exchange 5.5 to Exchange 2000

When Exchange 2000 receives an inbound replication message from an Exchange 5.5 server the following properties are dropped:

ptagNTSD

ptagAdminNTSD

This prevents a security hole, where these entries could be modified on Exchange 5.5 and replicated back into Exchange 2000.

User DNs are replaced by SIDs

Exchange 2000 allows client access to folders based on the user's NT SID, not on the their mailbox DN. When an ACL replicates from Exchange 5.5 to Exchange 2000 the store will look up the DNs in the W2K AD and place the users' SIDs on the folders' ACL list. **This means that Exchange 2000 must be able to find the DN of the Exchange 5.5 user in the Active Directory and find the SID associated with the account. If it can't then the permissions for that folder will not upgraded properly and users may not be able to access it (or even see it).**

> ***More Information***
>
> A trusted NT4 domain is any domain that the W2K domain accesses via an old style NT4 trust. The domain itself could actually be a W2K domain in a separate forest, but W2K will essentially see it as though it were an NT4 domain.

Upgrade of ptagACLData to NTSD

When Exchange 2000 receives ptagACLData it must promote this information into ptagNTSD before the folder can be accessed.

This is done using temporary tables to extract the information from ptagACLData and promote the ACLs into ptagNTSD.

The store enumerates the ACL list and attempts to obtain the SID for each of the users (or groups) ACL'd on the folder. Once it completes, all the users/groups are promoted into ptagNTSD and the temporary ACL tables removed.

If any one DN fails for any reason (i.e. Master SID cannot be found) – the entire list of users fails to be promoted. This is to prevent a security hole where a user might be expressly denied permission, but the group they belong is allowed access. Also anonymous permissions will not be set.

Each time the folder is accessed (either by client or through replication), the store will re-attempt to promote the ptagACLData permissions to NTSD – until it is successful.

The permissions may look correct through a MAPI interface because the temp tables still exist. However, access in Exchange 2000 is only granted by the list of accounts in ptagNTSD.

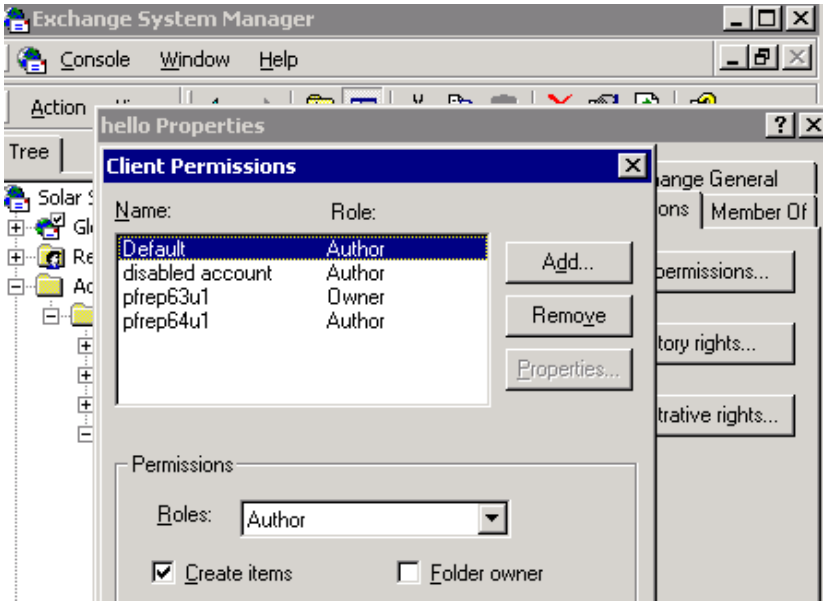
To view the actual NTSD permissions on a folder, hold down "CTRL" when clicking "Client permissions". This will display the raw NTSD ACL. This is what will govern access to the folder.

Example

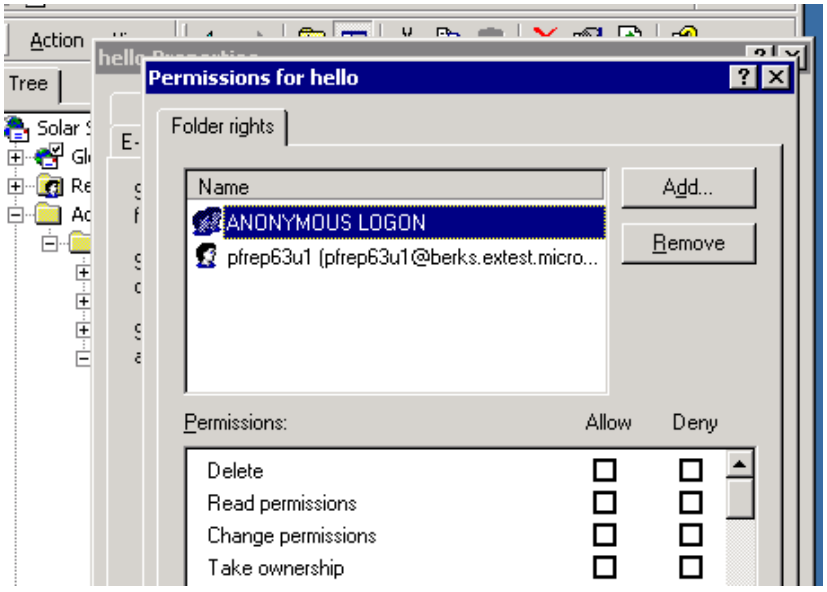
User “disabled account” is a disabled user with no Associated External Account therefore; there is no Master Account SID.

User pfrep64u1, is a valid W2K account, user pfrep63u1 is the owner.

Viewing the MAPI client permissions shows the following:



However, viewing the raw NTSD permissions shows the following



Neither user "Disabled User" nor "pfrep64u1" has access to the folder – even though there is nothing wrong with user "pfrep64u1"

Only pfrep63u1 is allowed access because they are the owner. However, even they cannot modify permissions on the ACL list until either "Disabled User" is given a Master Account SID, or they remove "Disabled User" from the ACL.

This can explain why MAPI permissions may look different from the NTSD permissions.

The following event will be logged:

```
Event 9548
Disabled user /O=SOLAR
SYSTEM/OU=MERCURY/CN=RECIPIENTS/CN=DISABLEDACCOUNT does not have a
master account SID. Please use Active Directory MMC to set an
active account as this user's master account.

And

Event 9551
An error occurred while upgrading the ACL on folder [Public
Folders]/hello located on database "First Storage Group\Public
Folder Store (PFREP63)".
 The Information Store was unable to convert the security for
/O=SOLAR SYSTEM/OU=MERCURY/CN=RECIPIENTS/CN=DISABLEDACCOUNT into a
Windows 2000 Security Identifier.
 It is possible that this is caused by latency in the Active
Directory Service, if so, wait until the user record is replicated
to the Active Directory and attempt to access the folder (it will
be upgraded in place).   If the specified object does NOT get
replicated to the Active Directory, use the Microsoft Exchange
System Manager or the Exchange Client to update the ACL on the
folder manually.
 The access rights in the ACE for this DN were 0x41b.
```

Scenarios

We will now look at several possible scenarios, and see what is ACL'd on a folder.

Scenario 1 – Exchange 5.5 user has a primary NT account in the W2K Domain

The User CA will replicate the Exchange 5.5 mailbox information to the user's account in the W2K AD.

The SID of the primary NT account on the Exchange 5.5 Mailbox will be used to ACL the folder.

Scenario 2 – Exchange 5.5 user has primary NT account in a trusted NT4 (or external W2K) domain.

A disabled user will be created in the W2K AD by the User CA, with an Associated External Account containing the NT4 SID.

As the user's account has an Associated External Account, that Associated External Account SID will be used to ACL the folder.

Scenario 3 – Exchange 2000 user has disabled W2K account and an Associated External Account (from a trusted domain).

This is really the same as Scenario 2 – except that the mailbox is on an Exchange 2000 server rather than an Exchange 5.5 server. Again the SID used to ACL a folder is from the trusted domain. (This is how the Exchange Team access mailboxes and folders in DogFood with our Redmond NT accounts.)

Scenario 4 – A USG is ACL'd on a Public Folder, and the members are W2K accounts

No problems exist here. The members of the USG will be able to access the folder based on the rights the USG has on the folder

Scenario 5 – A USG is ACL'd on a public folder, and the members are W2K disabled accounts.

This presents additional problems. The accounts added to the USG are the W2K disabled accounts. However, the user will be logging on with their Associated External Account. Ordinarily they would not be able to access folders because the USG does not contain details about their actual logon (NT4 accounts).

To make this work, W2K uses a process called Token Augmentation. When the user logs on, their access token is augmented with the SID of the USG that their disabled account belong to. This allows them access to folders ACL'd with a USG.

This is a W2K feature and not an Exchange feature. There is a fix in W2K SP1 to allow this to occur in Mixed Mode (in W2K RTM token augmentation only happening in W2K native mode). A pre-requisite for Exchange 2000 is W2K SP1.

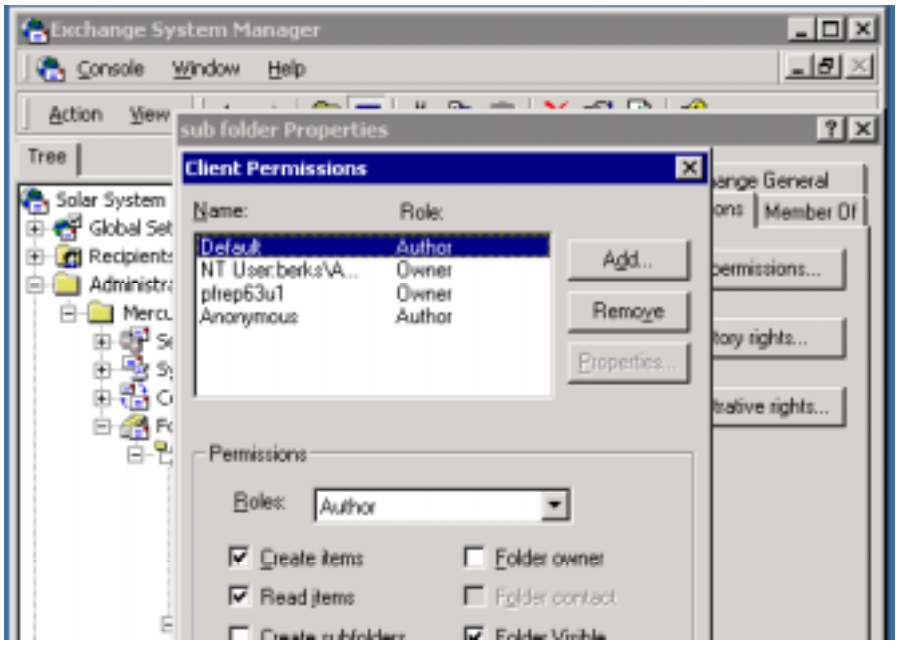
All of the above scenarios will work and users will have no problems accessing the folders.

Problems with Permissions

Windows 2000 User, who doesn't have a mailbox, is added ACL'd on folder

Cause: This happens when a user who doesn't have an Exchange mailbox creates or administers a folder either via ESM, or through IFS. The most likely scenario is that someone used an account that has permissions to administer folders (e.g. Enterprise Admin Account), but no mailbox has ever been created for that account.

Example: The domain admin account was used to create a new folder in ESM. Notice the strange user name "NT User: domain\<account name>"



Effect: Errors when trying to add users to the folder – "One or more users could not be added to the folder access list", either using a client or ESM. Default and anonymous permissions will not work. Only users who were previously ACL'd on the folder will be able to access it. Attempting to view properties of "bad" entry generates MAPI error.

Fix: Create a mailbox for the bad entry, or remove the bad entry from the ACL.

Comments: Strictly speaking this is not an Exchange 5.5 / Exchange 2000 interoperability issue. However, as this section is on permissions problems it's been included here. There is a fix in Exchange 2000 Service Pack 1, to allow the properties of the bad account to be viewed.

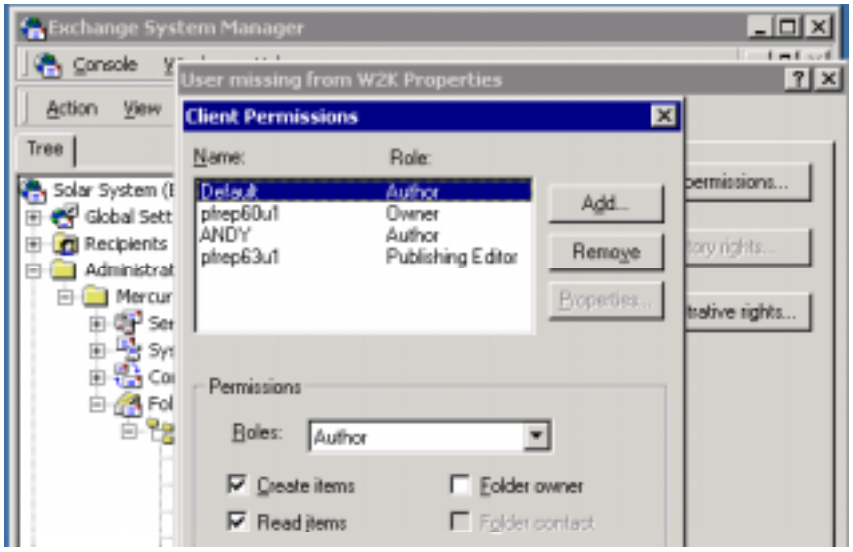
Exchange 5.5 user has not been replicated into W2K domain

Cause:

- No User CA in place to replicate the Exchange 5.5 mailboxes into W2K forest.
- Replication latency.
- User has been deleted from W2K domain.

When Exchange 2000 receives the replication message it will attempt to upgrade the data stored in ptagACLData to NT SIDs. If it fails, then only owners will be promoted to ptagNTSD. No one else will be able to access the folder.

Example: The Exchange 5.5 user Andy has been added to the ACL of a folder on Exchange 5.5. Exchange 2000 users will be unable to access the folder (or even view it).



Effect: Exchange 2000 users will have problems accessing the folder. Event 9551 will be logged every time the ACL attempts to upgrade (i.e. on client access or replication).

```
Event 9551

An error occurred while upgrading the ACL on folder [Public
Folders]/User missing from W2K located on database "First Storage
Group\Public Folder Store (PFREP63)".
 The Information Store was unable to convert the security for
/O=SOLAR SYSTEM/OU=MERCURY/CN=RECIPIENTS/CN=ANDY into a Windows
2000 Security Identifier.
 It is possible that this is caused by latency in the Active
Directory Service, if so, wait until the user record is replicated
to the Active Directory and attempt to access the folder (it will
be upgraded in place).   If the specified object does NOT get
replicated to the Active Directory, use the Microsoft Exchange
System Manager  or the Exchange Client to update the ACL on the
folder manually.
 The access rights in the ACE for this DN were 0x41b.
```

Fix: Either remove the bad entry or replicate the missing user to W2K.

Comments: If the folder is replicated to the Exchange 2000 server, the ACL will show the name in UPPERCASE, as DNs are always UPPERCASE. However, remember that ESM will connect to a store that holds an actual content replica of the folder to view the permissions. If ESM connects to an Exchange 5.5 server, then the ACL will "appear" correct. Do not fall into this trap. If the store is logging 9551 events, then these must be fixed before Exchange 2000 users can access the folder.

Disabled Account does not have a Master Account SID (Associated External Account).

Cause:

The W2K domain does not have a trust set up to the NT4 domain where the user's account is.

User has been manually disabled

Example: See the start of this section for an example of this.

Effect:

Only owners will be promoted to ptagNTSD. The permissions will actually look correct when viewed by ESM. However, viewing the raw NTSD permissions will show that only the owner has actually been added to the ptagNTSD ACL.

Fix:

- Remove the disabled accounts from the ACL
- Give the disabled accounts Associated External Accounts
- Create a trust between the NT4 (or external W2K domain) and the W2K domain the User CA is replicating to. This will give the disabled accounts Associated External Accounts (Master Account SIDs)

UDG cannot be converted to USG

Cause:

- UDG exists in Mixed Mode W2K domain
- MsExchDisableUDGConversion is enabled

Effect:

- Clients will get an error "The modified permissions could not be saved. The client operation failed"
- ESM will return, "The operation failed. ID no 8004005 Exchange System Manager."

In either case a 9556 event will logged, for example

```
Event 9556

Unable to set permission for DL /o=Solar
System/ou=Mercury/cn=Recipients/cn=group1 because it could not be
converted to a security group. This most likely is because your
system is in a mixed domain.
```

Summary

Public Folder replication between Exchange 5.5 and Exchange 2000 requires ADC CAs to be in place.

PF CAs are not used by replication – they merely allow folder's directory objects to be replicated between the Exchange 5.5 DS and W2K DC.

User CAs are vital to ensure permissions replicate properly.

Any single user's ACL entry that fails to be promoted into the new NTSD ACL can cause the entire ACL to fail to be upgraded, disallowing all user access to the folder. Additionally, anonymous permissions will not be set on the folder ACL.

UDGs will automatically be converted to USGs if the UDG is stored in a Native Mode W2K domain.

Check the event log for errors if users cannot access a folder.

- 9548 – couldn't find disabled user's Master Account SID.
- 9551 – user account could not be found in the Active Directory.
- 9552 – cannot upgrade UDG to USG.
- 9556 – cannot upgrade UDG to USG.

If there are permissions problems with a folder, try viewing the raw ACL by holding down CTRL when selected "Client permissions" (but do not actually use this to set the permissions, as this may cause more problems!).

Tip

It is strongly recommended that a DS/IS adjust for Public Folder permissions is run on Exchange 5.5 – just to remove any unknown accounts before introducing Exchange 2000 into the Organization. Only select the permissions option, do not start rehoming folders!

Emailing a Mail Enabled Public Folder

Sending an email to a folder is more complicated than sending an email to a mailbox. A mailbox can only exist on one server and is therefore anchored to a particular mailbox store. The directory entry for a mailbox points to a specific server so once the directory entry has been resolved, transport can use DN routing to work out which mailbox store to deliver the message to.

A public folder's directory entry has no "home" server. A public folder can exist on multiple servers and there is no information held in the directory to indicate which servers' actually holds replicas of the folder. That information is held in the store in the ptagReplicaList attribute. So the first priority of transport is to deliver the message to a store that knows where the replicas are. The store then looks up the ptagReplicaList entry for that folder and resubmits the message to transport, readdressed to a store that holds a replica of the folder.

Public Folder Directory Entry

LDP dump of Public Folder's directory entry.

```
>> Dn: CN=utils,CN=Microsoft Exchange System
Objects,DC=cumbria,DC=extest,DC=microsoft,DC=com
       1> homeMDB: CN=Public Folders,CN=Folder
Hierarchies,CN=Coniston,CN=Administrative Groups,CN=Lake
District,CN=Microsoft
Exchange,CN=Services,CN=Configuration,DC=cumbria,DC=extest,DC=micro
soft,DC=com;
       1> cn: utils;
       1> displayName: utils;
       1> mail: utils@Coniston.LakeDistrict.com;
       1> instanceType: 4;
       1> legacyExchangeDN: /o=Lake
District/ou=Coniston/cn=Recipients/cn=UTILS975FA60E47F93A7F283A2EB4
E18C94B900278A;
       1> distinguishedName: CN=utils,CN=Microsoft Exchange System
Objects,DC=cumbria,DC=extest,DC=microsoft,DC=com;
       1> objectCategory: CN=ms-Exch-Public-
Folder,CN=Schema,CN=Configuration,DC=cumbria,DC=extest,DC=microsoft
,DC=com;
       2> objectClass: top; publicFolder;
       1> objectGUID: e37e58cc-e1fc-435d-9463-c484ebf352f3;
       2> proxyAddresses: SMTP:utils@Coniston.LakeDistrict.com;
X400:c=US;a= ;p=Lake District;o=Coniston;s=utils;;
       1> name: utils;
       1> showInAdvancedViewOnly: TRUE;
       1> textEncodedORAddress: c=US;a= ;p=Lake
District;o=Coniston;s=utils;;
       1> uSNChanged: 21057;
       1> uSNCreated: 20607;
       1> whenChanged: 5/26/2000 18:20:40 Pacific Standard Time
Pacific Daylight Time;
       1> whenCreated: 5/26/2000 18:13:26 Pacific Standard Time
Pacific Daylight Time;
       4> msExchADCGlobalNames: forest:o=Lake
District00000000EE7B7DC579C7BF01;
EX5:cn=UTILS975FA60E47F93A7F283A2EB4E18C94B900278A,cn=Recipients,ou
=Coniston,o=Lake District:person$public-
folder$top00000000EE7B7DC579C7BF01;
FOREST:32149C43098F9243BDAFAAB9846D15F3000000003C55A80A79C7BF01;
NT5:CC587EE3FCE15D439463C484EBF352F3000000003C55A80A79C7BF01;
       1> folderPathname: utils;
       1> msExchHideFromAddressLists: TRUE;
       1> mailNickname: utils;
       1> replicatedObjectVersion: 28;
       1> replicationSignature: <ldp: Binary blob>;
       1> targetAddress:
expf:UTILS975FA60E47F93A7F283A2EB4E18C94B900278A;
       1> msExchMailboxSecurityDescriptor: <ldp: Binary blob>;
       1> deliveryMechanism: 0;
       1> dLMemDefault: 1;
       1> msExchPFTreeType: 1;
       1> msExchALObjectVersion: 22;
       1> msExchPoliciesIncluded: {CB137506-78E1-4583-BB76-
9F69D57DFAAE},{26491CFC-9E50-4857-861B-0CB8DF22B5D7};
```

How it works

The Categorizer (phatcat.dll) is responsible for correctly resolving the address of a message to a DN entry. In the case of public folders, it is also responsible to correctly determining which TLH the folder belongs to and addressing the message correctly to be submitted to a store in that TLH. It is then responsible for re-addressing the message to a store that holds a replica of that folder, once the replica list has been obtained.

Initial Folder Directory Entry Lookup

When a message is submitted to transport (either locally or from an external source), transport resolves the address to an entry in the Directory. If that entry is a folder (as opposed to a mailbox), the Categorizer must to obtain the HomeMDB of the folder.

```
1> homeMDB: CN=Public Folders,CN=Folder
Hierarchies,CN=Coniston,CN=Administrative Groups,CN=Lake
District,CN=Microsoft
Exchange,CN=Services,CN=Configuration,DC=cumbria,DC=extest,DC=micro
soft,DC=com;
```

A folder's HomeMDB contains the DN of the TLH this folder belongs to.

Note

HomeMDB has changed its meaning from Exchange 5.5. In Exchange 5.5 folders had a HomeMDB and a HomeMTA attribute that meant the MTA could easily determine which server to send the message to. Exchange 2000 can no longer rely on this. The MTA may not touch the message, and there can be many different hierarchies and stores. Folders in Exchange 2000 are no longer "Homed" on a particular server as they were in Exchange 5.5 and the ADC does not map HomeMDB in the Active Directory to Home MDB in the ExDS, nor does it replicate the HomeMTA data.

TLH server

Next the Categorizer looks up the TLH DN retrieved from the folder's HomeMDB attribute to obtain a list of all the servers in that folder's TLH. The Categorizer still doesn't know where the replica exists, but it can submit the message to a store that does.

The TLH DN contains a Backlink to all the servers in that TLH.

```
>> Dn: CN=Public Folders,CN=Folder
Hierarchies,CN=Coniston,CN=Administrative Groups,CN=Lake
District,CN=Microsoft
Exchange,CN=Services,CN=Configuration,DC=cumbria,DC=extest,DC=microsoft,
DC=com
       3> msExchOwningPFTreeBL: CN=Public Information Store
(PFREP55),CN=First Storage
Group,CN=InformationStore,CN=PFREP55,CN=Servers,CN=Windermere,CN=Adminis
trative Groups,CN=Lake District,CN=Microsoft
Exchange,CN=Services,CN=Configuration,DC=cumbria,DC=extest,DC=microsoft,
DC=com; CN=Public Folder Store (PFREP57),CN=First Storage
Group,CN=InformationStore,CN=PFREP57,CN=Servers,CN=Coniston,CN=Administr
ative Groups,CN=Lake District,CN=Microsoft
Exchange,CN=Services,CN=Configuration,DC=cumbria,DC=extest,DC=microsoft,
DC=com; CN=Public Information Store (PFREP56),CN=First Storage
Group,CN=InformationStore,CN=PFREP56,CN=Servers,CN=Coniston,CN=Administr
ative Groups,CN=Lake District,CN=Microsoft
Exchange,CN=Services,CN=Configuration,DC=cumbria,DC=extest,DC=microsoft,
DC=com;
       1> adminDisplayName: Public Folders;
       1> cn: Public Folders;
       5> dSCorePropagationData: 20000526214406.0Z; 20000526214404.0Z;
20000526210149.0Z; 20000526210028.0Z; 16010714223649.0Z;
       1> instanceType: 4;
       1> distinguishedName: CN=Public Folders,CN=Folder
Hierarchies,CN=Coniston,CN=Administrative Groups,CN=Lake
District,CN=Microsoft
Exchange,CN=Services,CN=Configuration,DC=cumbria,DC=extest,DC=microsoft,
DC=com;
       1> objectCategory: CN=ms-Exch-PF-
Tree,CN=Schema,CN=Configuration,DC=cumbria,DC=extest,DC=microsoft,DC=com
;
       2> objectClass: top; msExchPFTree;
       1> objectGUID: 892d18b0-123e-4090-9e38-ff3851038cfd;
       1> name: Public Folders;
       1> showInAdvancedViewOnly: TRUE;
       1> systemFlags: 1610612736;
       1> uSNChanged: 16047;
       1> uSNCreated: 3748;
       1> whenChanged: 5/26/2000 14:0:25 Pacific Standard Time Pacific
Daylight Time;
       1> whenCreated: 5/26/2000 12:58:11 Pacific Standard Time Pacific
Daylight Time;
       1> msExchPFTreeType: 1;
       1> msExchPFDefaultAdminACL: <ldp: Binary blob>;
```

The Categorizer then chooses a server from the msExchOwningPFTreeBL to send the message to. Rather than call routing (which would be CPU expensive) it uses the following algorithm (which can be calculated without calling routing).

Closest Store	Comments
1. Local store	Is there a store in the TLH that is on the local server? In which case re-address and deliver locally.
2. Exchange 2000 Store in the same Routing Group	Are any of the stores on servers in the same RG? If more than one then load balance between them.
3. Exchange 2000 Store in the same Admin Group	Are any of the stores on servers in the same AG? If more than one then load balance between them.
4. Exchange 5.5 Store in the same Admin Group / Site*	This will only apply to the MAPI TLH. If there are no Exchange 2000 servers in the local site with MAPI TLH stores, then we will send the message to an Exchange 5.5 server.
5. The first Exchange 2000 server in the msExchOwingPFTreeBL list	If there aren't any stores in the local admin group that are members of the TLH, then send to the first Exchange 2000 entry in the TLH list. This will be the most recently added Exchange 2000 store in the TLH. This is the last resort for an App TLH.
6. The first Exchange 5.5 server in the msExchOwingPFTreeBL list*	MAPI TLH only. This is an extremely unlikely scenario, which will only happen when the Exchange Organization's directory is in a state of flux. In Exchange 2000 mixed mode (i.e. coexistence with 5.5), you must have at least one MAPI store per AG. However, it is possible that details about local MAPI stores may not have replicated yet, so as a last resort we will try an Exchange 5.5

Note:

*We try to avoid using an Exchange 5.5 store. This is due to how the store determines the location of the "best" replica to re-direct the message onto. The Exchange 5.5 store will use the same process for calculating the "best" replica as it would for re-directing a client lookup. If the replica(s) were external to the site and the Exchange 5.5 server had no affinities to that site(s), it would be unable to "see" a store with a replica, and NDR the message. This is OK in a pure Exchange 5.5 topology because the site that contained the server referenced by the ExDS HomeMTA & HomeMDB attributes, *by definition*, had to contain a replica of the folder. For more information see **Public Folder Referral and Public Folder Affinity**.

Addressing

The Categorizer then addresses the message to the chosen store, and the message is sent.

Example

This is a message sent by user “pfrep56u1” on server PFRFEP56 to a MAPI folder called “Support Issues” which only has a replica on server PFREP57.

Before it has been handled by the categorizer, there is no information about where to route the message to.

```
Sender Information:
               SMTP: NONE
               X500: NONE
               X400: NONE
               LegacyEXDN: /O=LAKE DISTRICT/OU=WINDERMERE/CN=RECIPIENTS/CN=PFREP56U1
               Other: NONE
               Msg822Subject: this is a mail to a public folder located on another
server
               InternetMsgID:
<4E18B371E0E111418E5A5926B0F0F7B305B012@pfrep56.cumbria.extest.microsoft.com>
               MTS_ID: c=US;a= ;p=Lake District;l=PFREP56-000907193034Z-1
               EMP MsgClass: IPM.Note


       Recipient #1
               SMTP: NONE
               X500: NONE
               X400: NONE
               LegacyEXDN: /O=LAKE DISTRICT/OU=Grasmere/cn=Recipients/cn=SUPPORT
ISSUESCC53D21BCC53D21BCC53D21B1EF3622B00177D
               Other: NONE
```

The categorizer determines that the message should be sent to itself (as it has a copy of this TLH locally).

This is the message after the categorizer has handled it

```
       Sender Information:
              SMTP: pfrep56u1@Windermere.LakeDistrict.com
              X500: CN=pfrep56u1,CN=Users,DC=cumbria,DC=extest,DC=microsoft,DC=com
              X400: c=US;a= ;p=Lake District;o=Windermere;s=pfrep56u1;
              LegacyEXDN: /o=Lake District/ou=Windermere/cn=Recipients/cn=pfrep56u1
              Other: NONE
              Msg822Subject: this is a mail to a public folder located on another
server
              InternetMsgID:
<4E18B371E0E111418E5A5926B0F0F7B305B012@pfrep56.cumbria.extest.microsoft.com>
              MTS_ID: c=US;a= ;p=Lake District;l=PFREP56-000907193034Z-1
              EMP MsgClass: IPM.Note

       Recipient #1
              SMTP: supportissues@Grasmere.LakeDistrict.com
              X500: CN=support issues,CN=Microsoft Exchange System
Objects,DC=cumbria,DC=extest,DC=microsoft,DC=com
              X400: c=US;a= ;p=Lake District;o=Grasmere;s=support issues;
              LegacyEXDN: /o=Lake District/ou=Grasmere/cn=Recipients/cn=SUPPORT
ISSUESCC53D21BCC53D21BCC53D21B1EF3622B00177D
              Other: NONE
              MDB guid: {B8E20FA8-9826-43E2-A969-C0D7C7B9B964}
              RP_DOMAIN: Windermere.LakeDistrict.com
```

The categorizer has determined that this message should be delivered to the store with MDB GUID B8E20FA8-9826-43E2-A969-C0D7C7B9B964 – which in this case is itself, PFREP56.

Note

The actual address used on the email will be formatted as required depending upon the actual transport used to transfer the message.

Example

For an MTA the address is of the form:

<DN of Store> /DDA: <LegDN of Public Folder>

Choosing the Content Replica

Transport will then attempt to deliver the message to the store.

Note

At this point the trace information will be removed from the message

The store will read the recipient details from the message.

The store uses almost the same method to determine the nearest store as it would for a client folder referral.

Steps	Comments
1. Look up the folder entry in the hierarchy and obtain a list of the Owning MDBs	
2. Checks to see if there is a local replica	In which case the message will be delivered locally.
3. Checks to see if there is a replica in the same RG	If there is then the store with a replica is returned to transport. If there is more than one it will load balance.
4. If there is no replica in the same RG then the list of stores will be sorted by cost.	The list is sorted by cost. The store calls routing to determine the "cost" to get to each server, then qsorts the list. The store caches the cost to a store for 1 hour, to reduce the number of calls it has to make to Routing. (In Exchange 5.5 the store used the affinity table to sort the list)
5. The cheapest store is chosen.	If more than one store is cheapest, then it will load balance. When calculating the cost, routing ignores the "Disallow PF Referral" setting on connectors, as this is not a client referral, but will consider the link state to the store.
6. If no replica can be found, then the message will be NDR'd	This can occur if PF replication has not replicated the folder's hierarchy entry to this store, due to replication latency

In summary there are three possible outcomes to this process.

- The store accepts the message and delivers it to the folder
- The store returns an alternate store that has a replica
- The store returns an error because it can't find the folder. In this case the message will be NDR'd back to the sender.

Re-addressing

If the store returns an alternate server to deliver to, the Categorizer will re-address the message to the new store and the process will repeat.

Example

Continuing on from the previous example, the message is returned the categorizer.

```
        Sender Information:
                SMTP: pfrep56u1@Windermere.LakeDistrict.com
                X500: CN=pfrep56u1,CN=Users,DC=cumbria,DC=extest,DC=microsoft,DC=com
                X400: c=US;a= ;p=Lake District;o=Windermere;s=pfrep56u1;
                LegacyEXDN: /o=Lake District/ou=Windermere/cn=Recipients/cn=pfrep56u1
                Other: NONE
                Msg822Subject: this is a mail to a public folder located on another
server
                InternetMsgID:
<4E18B371E0E111418E5A5926B0F0F7B305B012@pfrep56.cumbria.extest.microsoft.com>
                MTS_ID: c=US;a= ;p=Lake District;l=PFREP56-000907193034Z-1
                EMP MsgClass: IPM.Note

        Recipient #1
                SMTP: supportissues@Grasmere.LakeDistrict.com
                X500: CN=support issues,CN=Microsoft Exchange System
Objects,DC=cumbria,DC=extest,DC=microsoft,DC=com
                X400: c=US;a= ;p=Lake District;o=Grasmere;s=support issues;
                LegacyEXDN: /o=Lake District/ou=Grasmere/cn=Recipients/cn=SUPPORT
ISSUESCC53D21BCC53D21BCC53D21B1EF3622B00177D
                Other: NONE
                MDB guid: {B8E20FA8-9826-43E2-A969-C0D7C7B9B964}
                RP_DOMAIN: Windermere.LakeDistrict.com
```

Based on the information about where the actual replica exists returned by the store. The categorizer re-categorizes the message:

```
        Sender Information:
                SMTP: pfrep56u1@Windermere.LakeDistrict.com
                X500: CN=pfrep56u1,CN=Users,DC=cumbria,DC=extest,DC=microsoft,DC=com
                X400: c=US;a= ;p=Lake District;o=Windermere;s=pfrep56u1;
                LegacyEXDN: /o=Lake District/ou=Windermere/cn=Recipients/cn=pfrep56u1
                Other: NONE
                Msg822Subject: this is a mail to a public folder located on another
server
                InternetMsgID:
<4E18B371E0E111418E5A5926B0F0F7B305B012@pfrep56.cumbria.extest.microsoft.com>
                MTS_ID: c=US;a= ;p=Lake District;l=PFREP56-000907193034Z-1
                EMP MsgClass: IPM.Note

        Recipient #1
                SMTP: supportissues@Grasmere.LakeDistrict.com
                X500: CN=support issues,CN=Microsoft Exchange System
Objects,DC=cumbria,DC=extest,DC=microsoft,DC=com
                X400: c=US;a= ;p=Lake District;o=Grasmere;s=support issues;
                LegacyEXDN: /o=Lake District/ou=Grasmere/cn=Recipients/cn=SUPPORT
ISSUESCC53D21BCC53D21BCC53D21B1EF3622B00177D
                Other: NONE
                MDB guid: {4758C926-5CA5-4BA4-AC3A-58A536E9FB58}
                P_DOMAIN: pfrep57.cumbria.extest.microsoft.com
```

The message will now be routed to the store with MDB GUID 4758C926-5CA5-4BA4-AC3A-58A536E9FB58, which is PFREP57 and actually has the replica of the folder.

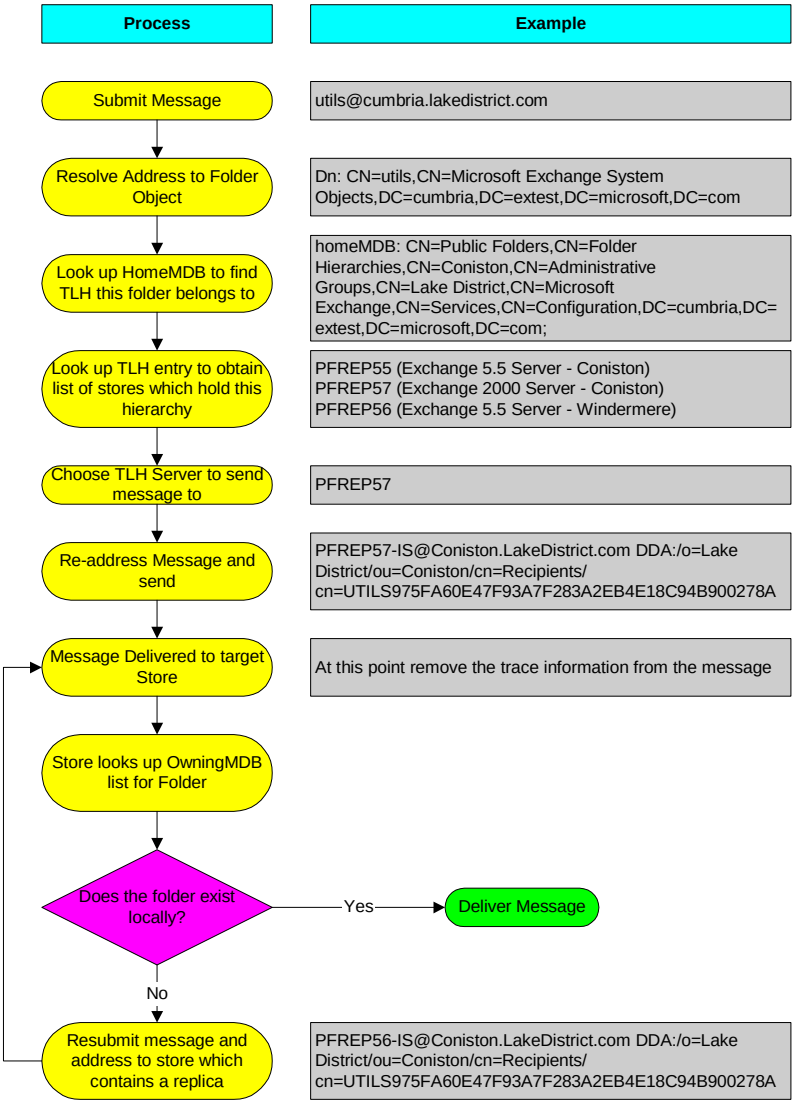
If for some reason this new store didn't have a replica (due to the original store having the wrong information due to replication latency), the process will repeat.

Important

Every time transport submits a message to the store the trace information is removed from the message. This is to prevent incorrect loop detection of the message. An email to a public folder may have to return over a connector it has already been sent across before, due to the location of the message's final destination.

Summary of Emailing a Public Folder

This is a flow chart showing how this works in practice.



Specific problems with a mixed Exchange 2000 /Exchange 5.5 topology

Mailing Application TLH folder

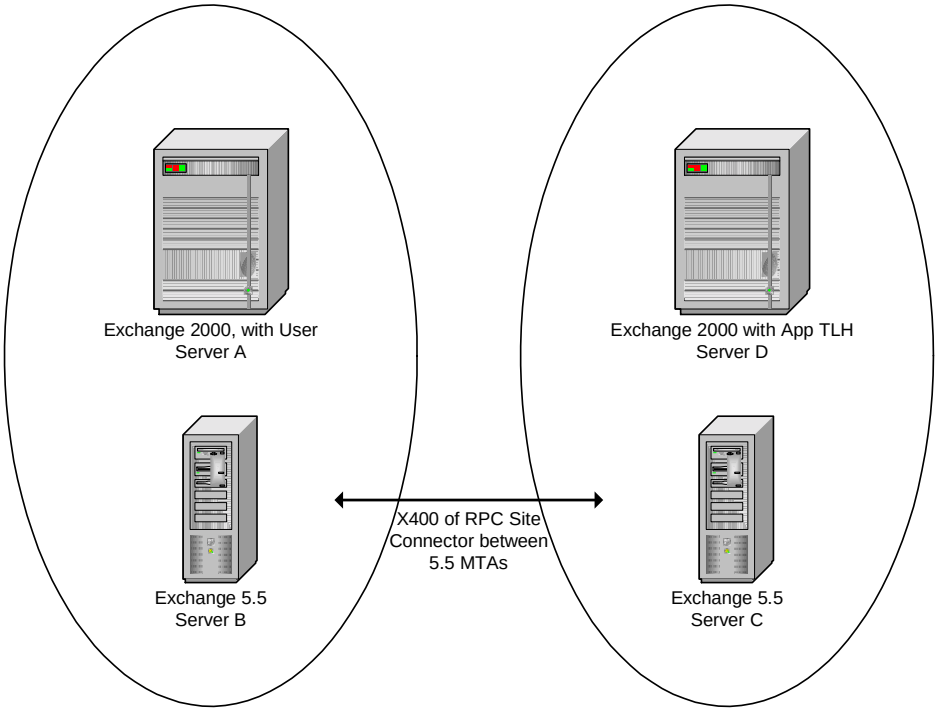
Email cannot be sent to App TLH public folders, if an Exchange 5.5 MTA ever resolves the DN of the target App TLH store.

This is a limitation caused by the way we allow App TLH public folder replication via Exchange 5.5 servers.

When the Exchange 5.5 MTA looks up the address of the App TLH store, it will find a match on the "pilgrimmed" X.500 entry. It will then re-address the email using the Obj-dist-name attribute for the value of the DN. This loses the DDA value that contains the name of receiving folder (as the Exchange 5.5 MTA will not preserve DDA values when rewriting the email address).

The message will be NDR'd

Example



Explanation

User on Server A attempts to email an App TLH folder on Server D.

The email will be NDR'd by Server D, because the MTA on Server C has removed the original recipient address and replaced it with the value from the Exchange 5.5 directory, thereby losing the DDA value which indicates which folder the mail is for

1. The message arrives at Server C with the correct OR address of: **DN:/o=Org/ou=Site/cn=Configuration/cn=Servers/cn=Server D /DDA: o=Org/ou=Site/cn=Recipients/cn=<foldername+GUID>**

2. The MTA on Server C sees that the DN matches it's own site, and so looks in its directory for the Home-MTA attribute of the DN. It can't find it (because we don't replicate App TLH stores into the Exchange 5.5 directory as Public Stores), but it does find a "pilgrim" address on an object of: **X500:/o=Org/ou=Site/cn=Configuration/cn=Servers/cn=Server**

3. The MTA replaces the address with the Obj-Dist-Name of this object, so the address becomes: **/o=Org/ou=Site/cn=Configuration/cn=Exchange 2000 Objects/cn=App TLH Store**.

4. The DDA value is lost, so when the message arrives at Server D, it has no idea which folder this message is for, and the message NDRs.

More Information

For more information on why the X.500 pilgrim address is added to the Exchange 5.5 directory object, see **Exchange 5.5 and Exchange 2000 Folder Replication**.

Workarounds

- Replicate the folder to the local site where the users are mailing from
- Replicate the folder to a site that the message can be delivered to without the message touching an Exchange 5.5 MTA.
- Upgrade the links to Exchange 2000 connectors.

Transport and Routing

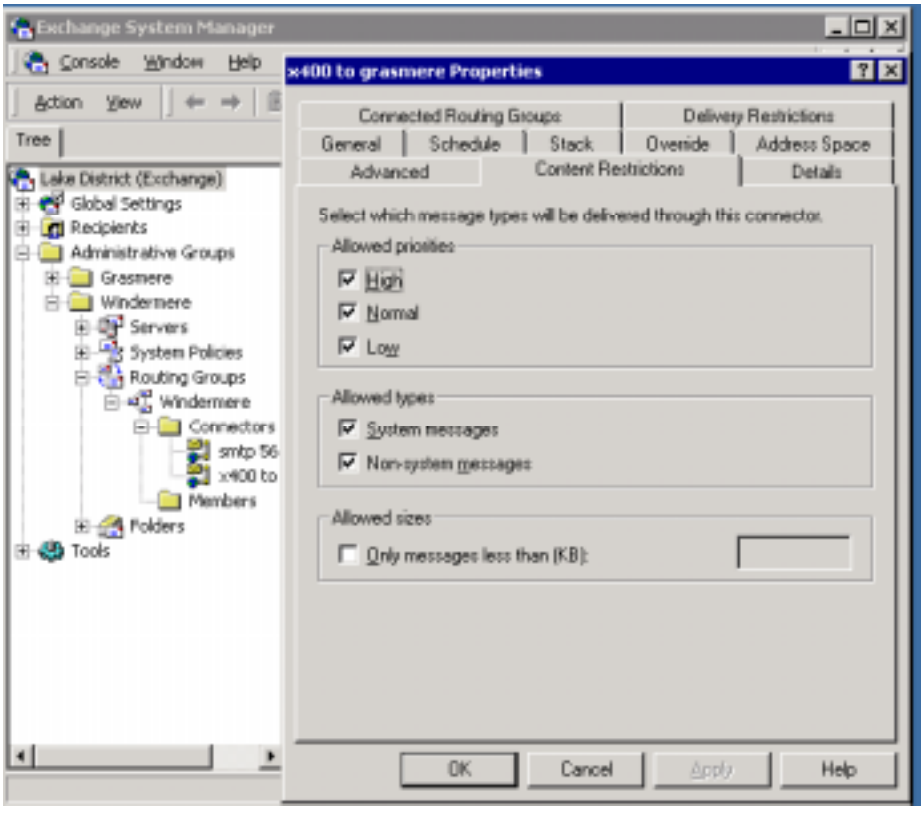
This section covers how various settings can affect Public Folder Replication messages.

Public Folder Replication Messages are **System Messages**

Allowing System Messages

In Exchange 5.5 it was possible to restrict delivery system and non-system messages over an X.400 connection (using Heuristics settings in the Exchange 5.5 directory). This facility has been extended for all replication transports in Exchange 2000:

- X.400 Connector
- SMTP Connector
- RG Connector



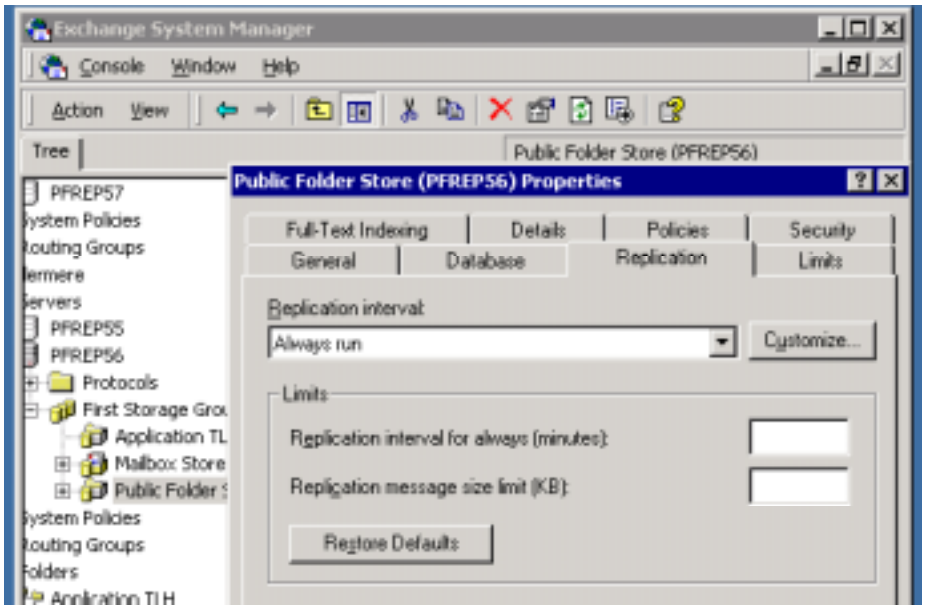
Unchecking “System messages” will prevent PF Replication Messages using this link.

Size Limits

Connector size restrictions **do not** affect system messages. This is because the replication engine cannot control the maximum size of a replication message and if system messages obeyed size restrictions then Public Folders could fall out of sync.

Replication Message Size Limits

The smallest unit of replication is an individual update to a folder. One of the most misunderstood settings in PF Replication is "Replication Message Size Limit" on the properties of a Public Folder Store.



Replication message size limit (KB) = 300KB by default. (The value is not shown unless explicitly set in Exchange 2000 RTM, this will be fixed in later versions).

This means that up to 300KB of different changes will be packed into a single replication message. It does not mean chop a 900KB update into 3 separate replication messages.

Example – (this is simplified to demonstrate the point)

I post 40 10KB updates into a public folder. The store will pack 30 updates into 1 replication message (300KB), and another 10 into a second replication message (10KB).

I post a single 5MB update into a public folder. The store will generate one 5MB replication message.

Preventing Large Replication Messages

To prevent large replication messages, limit the Maximum Posting allowed to a Public Folder.

Add a Rule to the Folder

Add a rule to automatically delete messages over a certain size. This will prevent large items being posted to the folder. This was the only way you could prevent large replication messages in Exchange 5.5

Prevent Large Messages being posted to a Folder

This option is only available when the Exchange 2000 Organization is in Native Mode. This can be used to restrict the maximum message size added to a Public Folder.

This is not supported in Mixed Mode, because Exchange 5.5 cannot handle this properly.

Delivery Restrictions

Delivery Restrictions **do not** affect System Messages.

Example

If a connector is configured to only "Accept messages from: specified users, this will not apply to Public Folder Replication messages. They will continue to use this connector

Priority Restrictions

Priority Restrictions **do** affect System Messages.

The replication message priority can be set for individual Public Folders, as a property on the folder. Replication messages will obey connector priority restrictions

Summary

Connector Setting	Affects PF Replication Message
Allow System Messages	Yes
Size Limits	No
Connector Delivery Restrictions	No
Priority Restrictions	Yes

Special Replication Cases

Search Folders

Search folders can be created via DAV to allow data from multiple folders that match the search criteria to be pulled into a single folder.

The search folder contains a search query and the store populates the folder with links to messages that match the query.

Search folders themselves do not contain messages (though it appears to clients that they do). Instead they hold a set of pointers (back links) to the actual messages.

The search query is performed locally and can only search folders that actually have content replicas on the store where the search is being performed – it cannot use referral to access data.

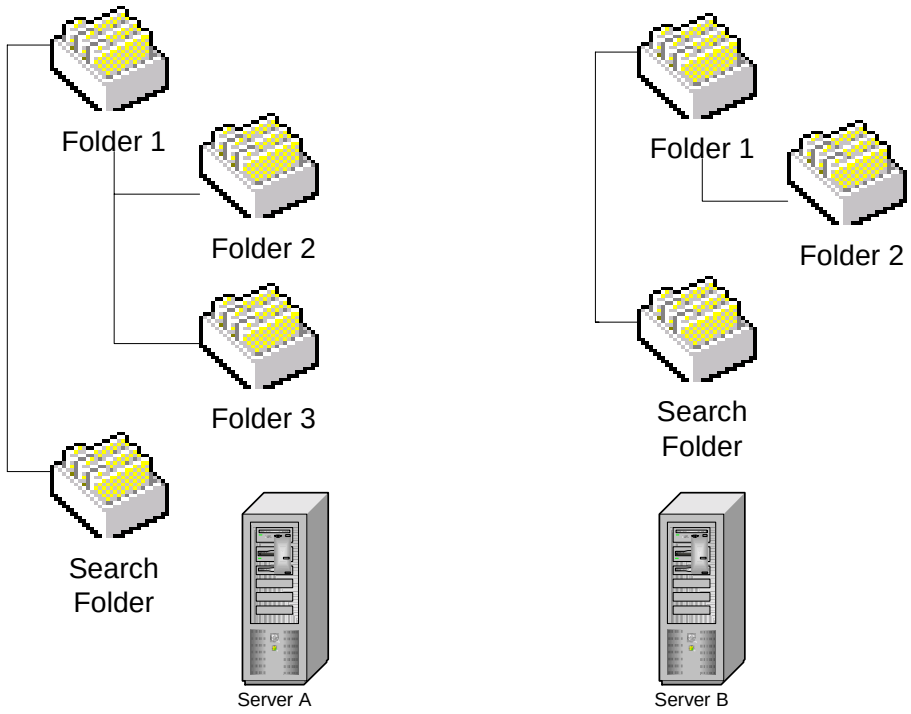
Search folders can be replicated so the search is done on other stores. However, when they replicate only the search query replicates. The results of the search do not replicate. Instead the search results are recalculated based on the search query.

Explanation

This diagram only shows content replicas on Server A and B. In this example Server A has replicas of Folders 1,2 & 3. Server B has replicas of Folders 1 & 2.

A search folder "Search" is created which queries Folders 1,2 & 3 and displays the results. This folder is also replicated to Server B.

Example



A search folder called "Search" has been created on Server A. It searches Folders 1,2 & 3 for certain criteria and displays the results.

The search folder is then replicated to Server B. Only the search query itself replicates – not the results of the search. As there is no content replica of Folder 3 on Server B, then it cannot search Folder 3 and will only display results that match the query from Folders 1 & 2.

If Folder 3 is replicated to Server B in the future, then the search will be redone and results from Folders 1,2 & 3 will be displayed in the search folder on Server B.

Recurring Appointments

Calendar data can be stored in Public Folders. Recurring appointments have to be handled slightly differently from other calendar data when they replicate. This is due to how a recurring appointment is stored for different clients.

When a MAPI client creates a recurring appointment, a single message is placed in the calendar folder to indicate the recurring appointment. When a MAPI client views the appointment, the client does the work *locally* of expanding the appointment so it appears on multiple dates.

The Web Client (OWA) cannot do this. When a web client accesses a recurring appointment, the appointment is expanded in the store into multiple separate appointments – called expansion messages. These expansion messages do not replicate – only the master recurring appointment replicates. If the recurring appointment is edited, then the expansion messages are deleted and the master is re-expanded.

MAPI clients can only see the master recurring appointment – they cannot view the expansion messages.

This is to prevent problems with co-existence with Exchange 5.5 and reduces overall replication traffic.

Users should not notice anything. Recurring appointments will view correctly in both MAPI and OWA clients and replication handles which updates to replicate.

Example

Server A and Server B both have copies of a calendar public folder.

User A (on Server A) uses a MAPI client to create an appointment that recurs 5 times. This creates a single “message” in the calendar folder on Server A.

The message replicates to Server B.

User B (on Server B) then views the appointment with Web client. The store expands the master message into individual appointments. So now there are 6 messages in the calendar folder on Server: 1 master and 5 expansion messages.

These expansion messages will never replicate to another server, only the master message.

Implied Restriction

Expansion messages are one example of messages that have an implied restriction. This means that the replication engine will ignore these changes and will not replicate them to another store.

Another example of implied restriction is message expiry. Deletes caused by expiration do not replicate, it is up to the individual public stores to expire messages in their local folders.

Public Folder Referral and Public Folder Affinity

This section briefly examines how Public Folder Referral works in Exchange 2000. It has replaced Public Folder Site Affinity that was used in Exchange 5.5.

Public Folder Site Affinity was the Exchange 5.5 mechanism used by MAPI clients in one site to view the contents of public folders in a remote site, without the need the public folder's content to be replicated to client's site. It was set up on a per site basis and was non-transitive. Clients made direct RPC connections to a server in the remote site to access the public folder content. Affinities are assigned costs to determine the order in which they are chosen.

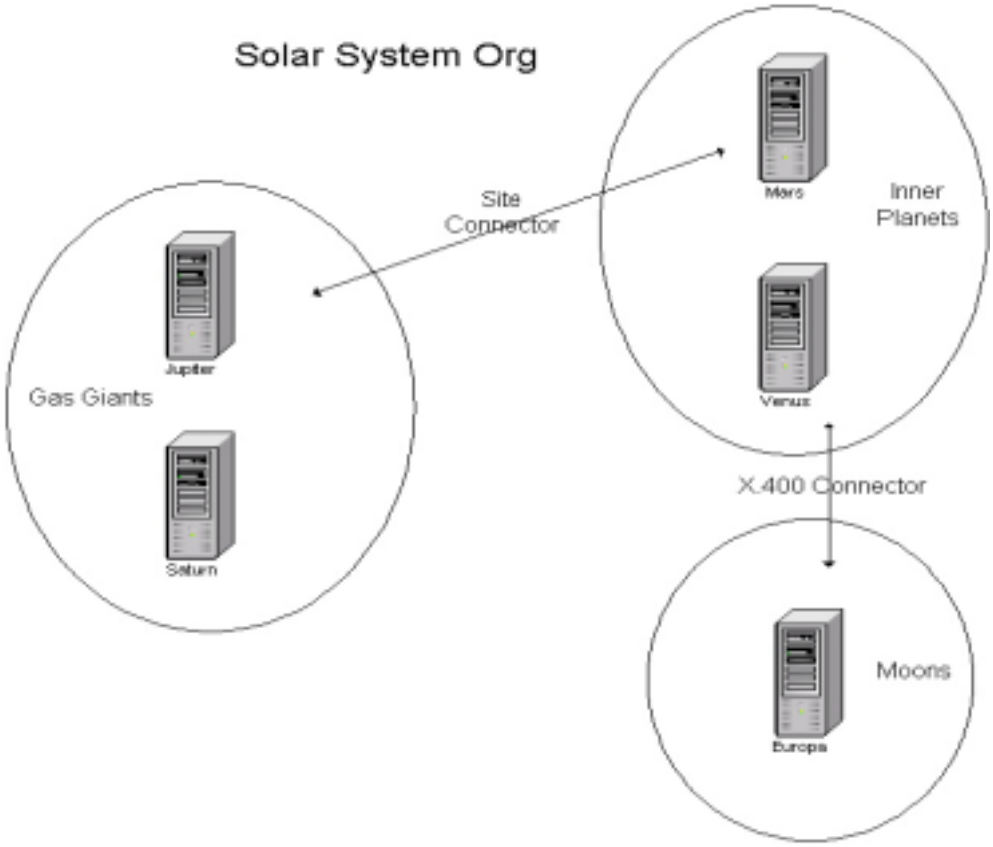
Public Folder Referral is the new Exchange 2000 mechanism by which MAPI and Web clients can view the contents of Public Folders in remote routing groups, without the need for the public folder's content to be replicated to the client's local routing group. Once again, the store simply tells the client where a replica is, and the client makes a direct connection to the server with the replica.

Note

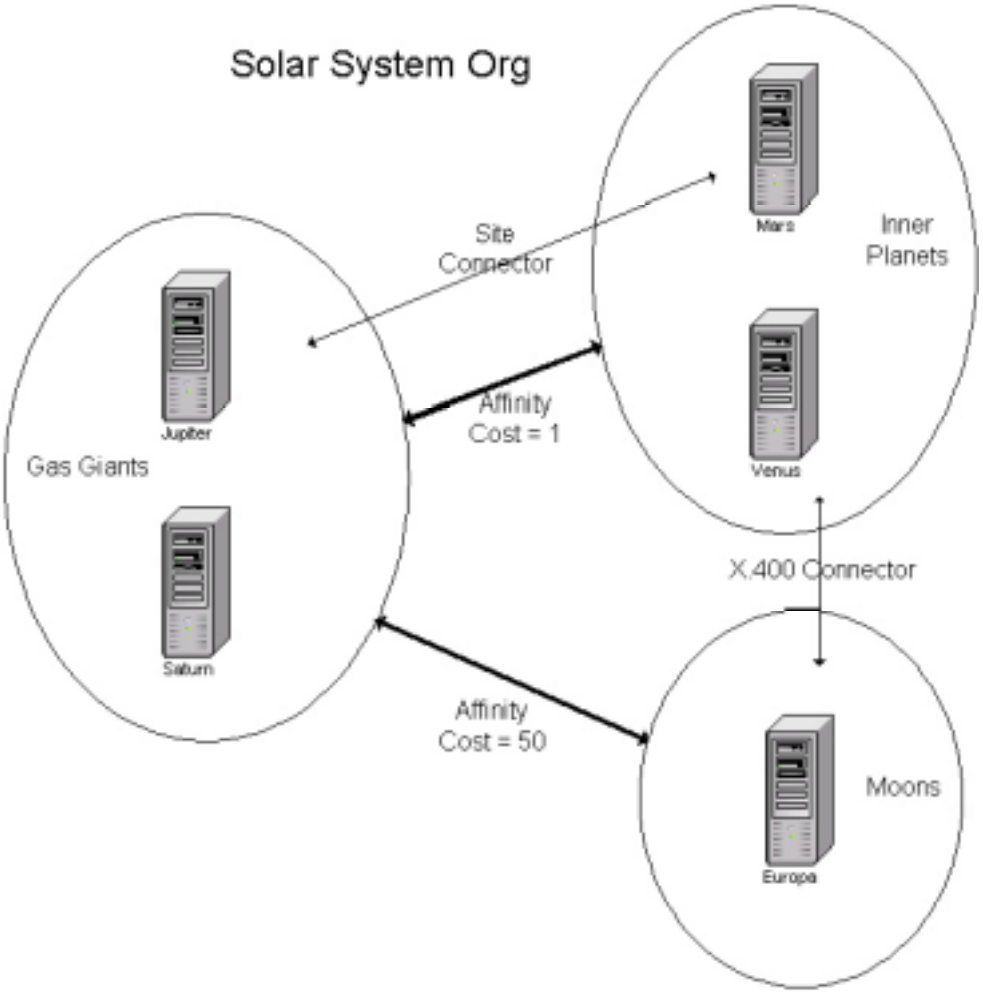
The Microsoft IMAP4 client does not support referral.

Recap on Public Folder Site Affinity

Below is an example of a simple Exchange 5.5 Organization. In this example of the Solar System Org, public folder content exists only in the "Inner Planets" site and the "Moons" site; the content does not exist in the "Gas Giants" site.



For users in the “Gas Giants” site to be able to access public folders’ contents replicated only in “Inner Planets” or “Moons”, Public Folder Site Affinities need to be created.



If a replica of a folder existed in both “Inner Planets” and “Moons’ then the one from “Inner Planets” would be chosen because it has the lowest cost.

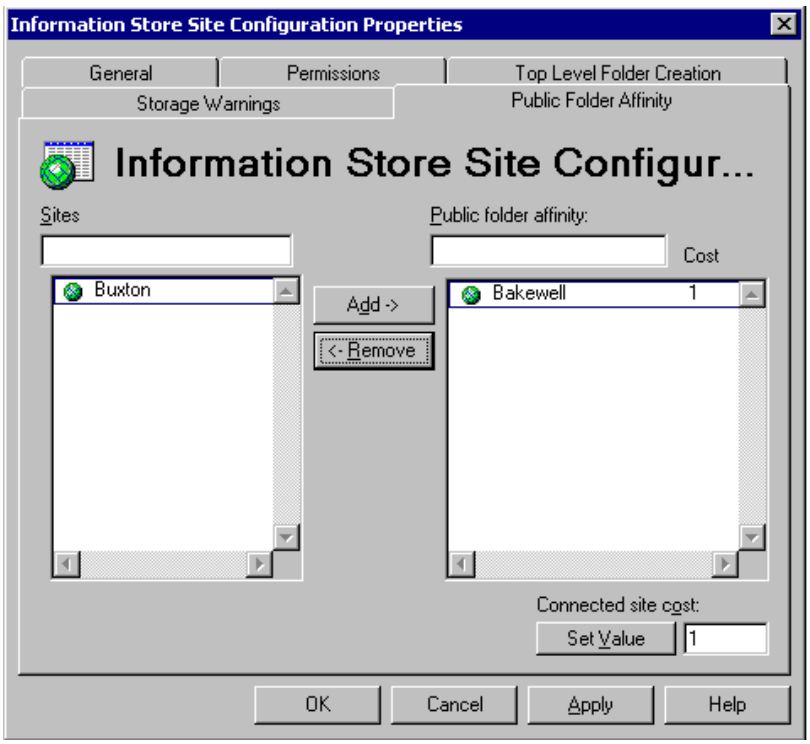
Affinities are Non-Transitive

In Exchange 5.5 Public Folder Affinities were non-transitive. If Site A has an affinity to Site B, which has an affinity to Site C, this does not mean that Site A has an affinity to Site C.

In the above example this means that “Moons” could not see public folder contents in “Inner Planets” without a separate affinity being created.

Creating Affinities

Public Folder Site Affinities are created using the Exchange 5.5 Admin program. They are a property of the **Information Store Site Configuration** object



In this example the site Bakewell can be used for affinity, but Buxton will not be used.

Choosing the Public Store

The user's public folder store (which may not necessarily be the same server as the user's mailbox) chooses which server to redirect the client to:

1. Client attempts to open a public folder.

2. Store.exe looks at the properties of that folder and retrieves the replica list, which lists which servers have a replica of the folder.

3. If the replica exists on the same server, then the client simply accesses the folder.

4. If a replica exists on a server in the same site* the client is sent to that server.

5. If there is no replica in the client's site, then store.exe sorts the list of servers based on their affinity costs and sends the client to the server with the lowest cost.

Note

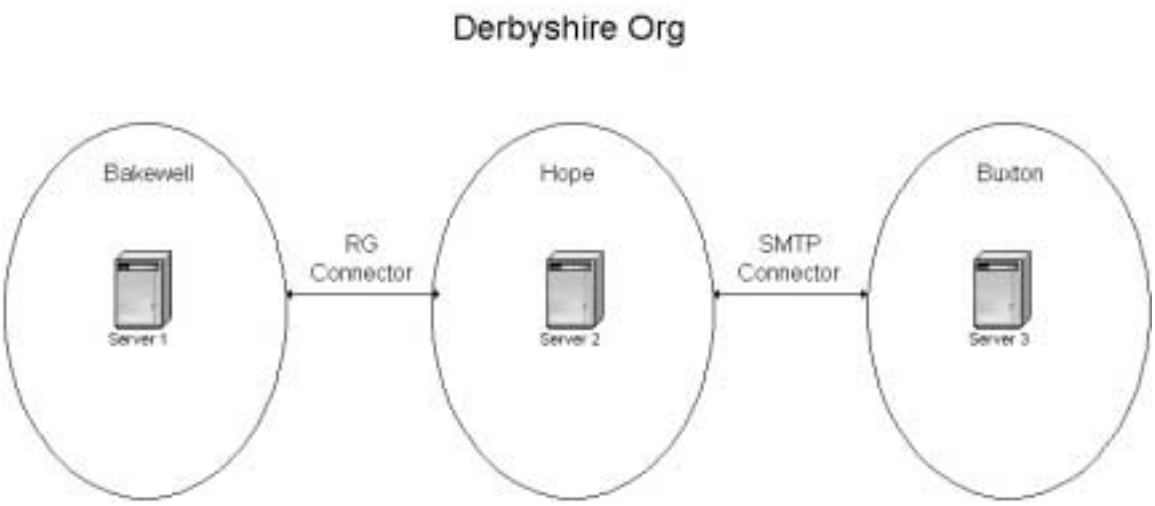
*The server chosen in the local site is not random. Each client will have a preferred server if multiple servers exist with the same replica. This helps to load balance across servers. Also Exchange 5.5 had the concept of **server locations**. A public folder replica in your own server's location would be chosen over a replica in a different location in the same site

Public Folder Referral

In Exchange 2000 a different mechanism is used to determine where clients are ***referred*** to in order to access public folder contents.

Folder referrals are now transitive, because the store uses the routing engine to determine if it can access a public folder

In the following topology there are 3 Exchange 2000 routing groups



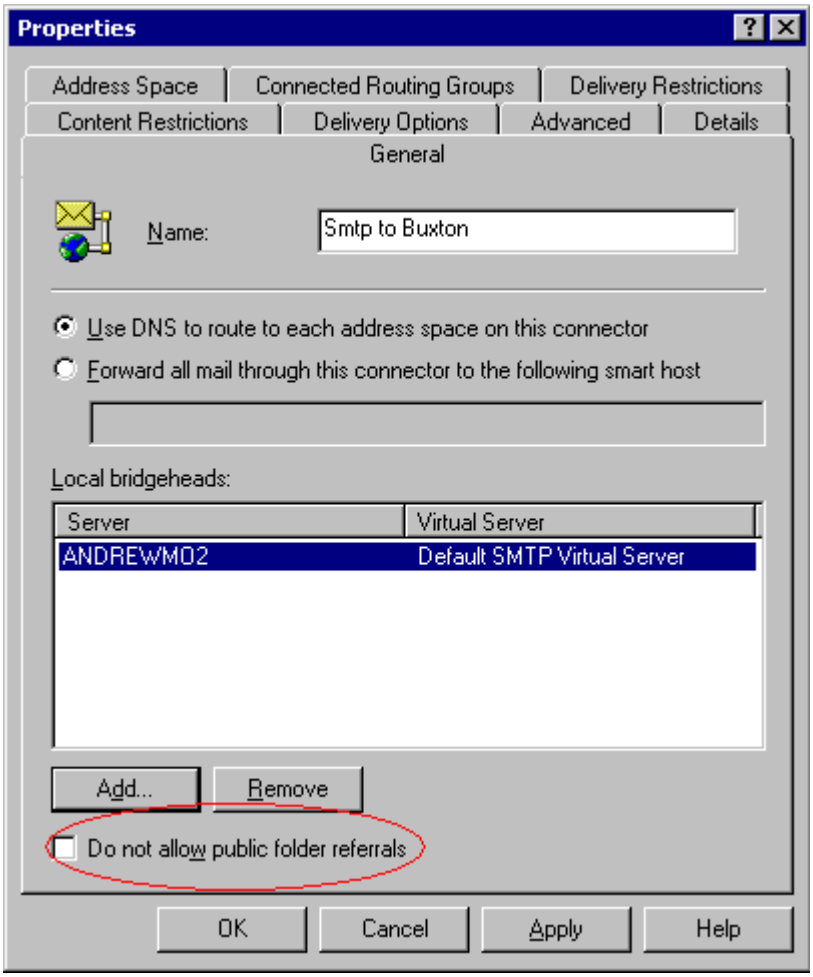
Both the RG Connector and the SMTP connector are set to **allow Public Folder referrals** (which is the default setting for Exchange 2000 connectors).

Exchange 2000 referrals are transitive. Therefore, users in Bakewell can access content in replicas on Server 2 in Hope and on Server 3 in Buxton.

Even if there were a second connector between Hope and Buxton that did not allow Public Folder Referrals, client referrals would still succeed because there is at least one path that does allow it.

Setting Referral Properties

A connector can be set to allow or disallow folder referrals in two places using ESM; on the properties of the connector, or by right clicking on a connector and toggling “Disallow Public Folder Referral”*.



Note

*As Exchange 5.5 connectors are read only in ESM, the only way you can control referral over an Exchange 5.5 connector is by right clicking on the connector and toggling the setting on the menu.

Choosing the Public Store

The user's public folder store (which may not necessarily be the same server as the user's mailbox) chooses which server to refer the client to:

1. Client attempts to view the contents of a folder.

2. Store.exe retrieves the replica list of the folder.

3. If the replica exists on the same server, then client simply accessed the folder.

4. If a replica exists on other servers in the same routing group, then the client is referred to one of these servers.

5. If the replicas only exist in other routing groups, then store.exe sorts the replica list by cost, by calling routing to look the cost to get to each server. The client is then referred to servers with the cheapest cost.

Sorting the Replica List

When the store sorts the replica list it takes account of whether a link has "Disallow Public Folder Referrals" set. If it is set then the cost becomes infinite and will not be returned to a client. Additionally if a Web Client requested the folder, then Exchange 5.5 servers are also removed from the list (as Exchange 2000 OWA cannot access an Exchange 5.5 server).

Additionally a client referral does not take into account routing **Link State Information**.

The costs returning from routing are cached by the store for 1 hour to reduce the number of times the store has to call into routing. This means that the effect of toggling "Disallow Public Folder Referrals" is not immediate.

More Information

If there are multiple servers in the same routing group that contain a replica, or multiple servers in remote routing groups with the same cost which contain a replica, then effectively Store.exe returns a choice of servers to the client (this is actually what happened in 5.5 as well). The client then chooses which server to connect to based on a random number the client assigns to that folder. This means that the client will always tend to go for the same folder. However the random number is different for other clients so they will always go for their own preferred server. This is design to meet two criteria.

- Achieve load balancing
- Clients will always see a consistent view of the folder.

Essentially the only difference between referral and affinity is that in Exchange 2000 the store uses routing to calculate the cost to a server, as opposed to using the costs in the affinity table.

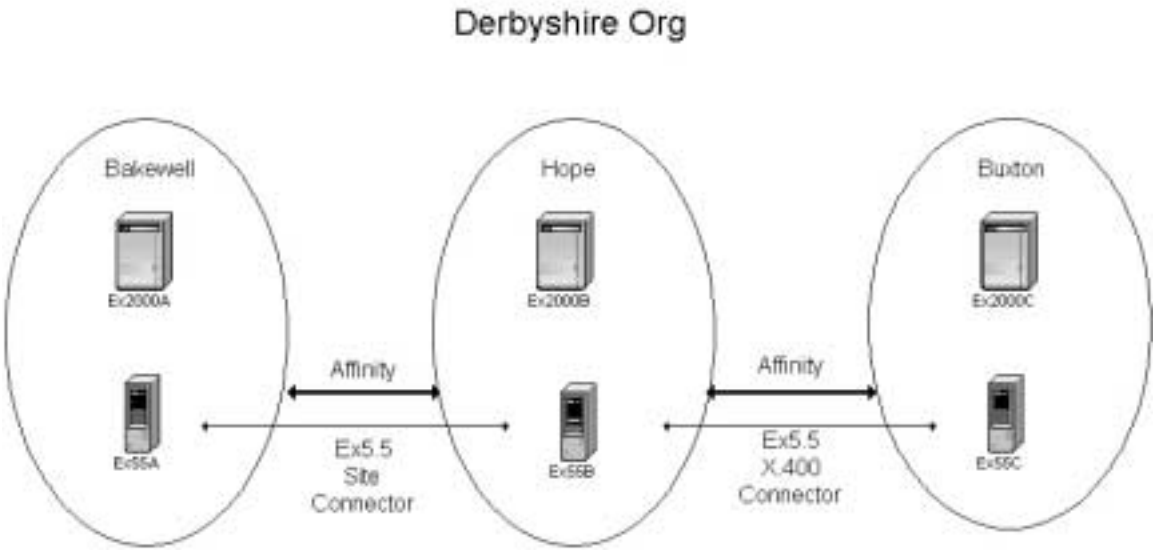
Mixed Exchange 5.5 and Exchange 2000 Organization

Exchange 2000 links allow Public Folder Referral by default.

Exchange 2000 views Exchange 5.5 links as having Public Folder Referral **disabled** by default. Because the property pages for 5.5 connectors in ESM are read only, you can enable an Exchange 5.5 connector via the right click menu.

The Config CA does not replicate Referral and Affinity information between W2K AD and Exchange 5.5. So users in mixed organizations will use whichever setting is appropriate for their version of Exchange. Exchange 2000 users will use referrals; Exchange 5.5 users will use affinities.

Example



In this topology there are no Exchange 2000 connectors set up. Therefore, even though Ex55A users can see the content of folders on Ex2000B and Ex55B, Ex2000A users will be unable to access the content by default.

To allow Ex2000A users access to public folders in Hope and Buxton, you would need to remove the “Disallow Public Folder Referral” setting from the 5.5 connectors in ESM.

Diagnostics, Event Logging & Tracing

Replication Issues

Turn up diagnostics logging on the Public Store and look at the replication events

Set the following diagnostics to Maximum:

- Replication Incoming
- Replication Outgoing
- Non-delivery reports

This will log replication messages being sent to and from the server, and also whether the replication message NDR'd

If there are specific problems with certain areas of replication, then additional logging may be required: Only set these once you've determined there's a replication problem.

- Replication Backfill
- Replication Errors
- Replication General

Permissions Issues

The following permissions will be logged with no diagnostics set:

Event ID	Description
9548	Disabled user %1 does not have a master account SID. Please use Active Directory MMC to set an active account as this user's master account.
9551	An error occurred while upgrading the ACL on folder %1 located on database "%4".%n The Information Store was unable to convert the security for %2 into a Windows 2000 Security Identifier.%n It is possible that this is caused by latency in the Active Directory Service, if so, wait until the user record is replicated to the Active Directory and attempt to access the folder (it will be upgraded in place). If the specified object does NOT get replicated to the Active Directory, use the Microsoft Exchange System Manager or the Exchange Client to update the ACL on the folder manually.%n The access rights in the ACE for this DN were %3.
9552	While processing public folder replication, moving user, or copying folders on database "%3", DL %1 could not be converted to a security group. Please grant or deny permissions to this DL on Folder %2 again. This most likely is because your system is in a mixed domain.
9556	Unable to set permission for DL %1 because it could not be converted to a security group. This most likely is because your system is in a mixed domain.

If you wish to view individual user's attempts to access folders set the following diagnostics to maximum:

- Logons
- Access Control

These will show permissions granted to users when they try to access folders.

Transport Issues

MTA

If the replication message is being delivered via an MTA (e.g. in mixed Exchange 5.5 and Exchange 2000 topology set the following diagnostics to maximum on the MTA:

- X.400 Service
- APDU – this will cause a rolling log of the Application Protocol Data Units to be written to the BF*.log files in exchsrvr\metadata.

Other Transports

Event logging for transports and routing can be set through diagnostics.

Also a logging feature called Regtrace is available, which can dump transport calls to a tracevwr (.atf) file, for analysis by PSS. Tracevwr.exe is available with SMS. This produces very detailed information and should only be used once the problem area has been narrowed down.

Regtrace

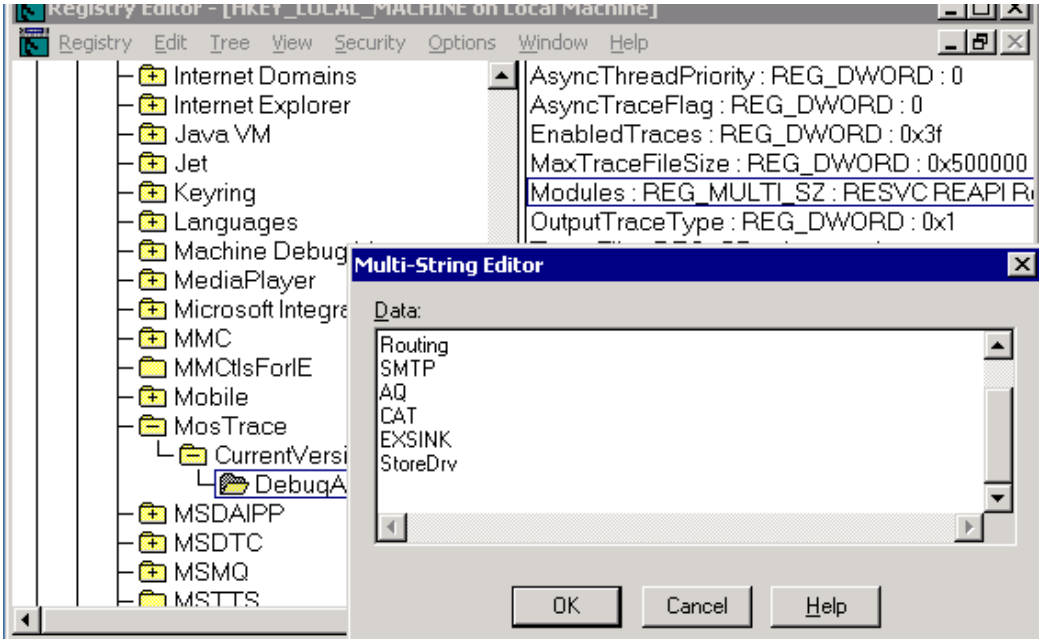
Using Regedt32.exe, in:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MosTrace\CurrentVersion\DebugAsyncTrace

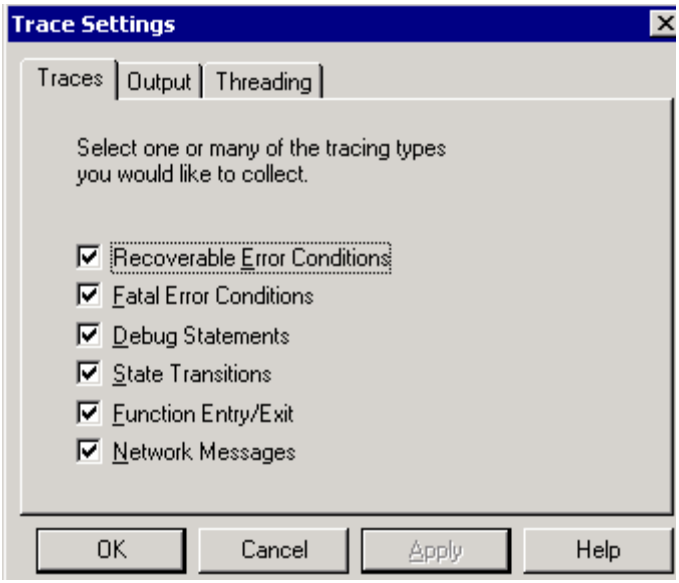
Create a new REG_MULTI_SZ value called:

Modules

Edit this value and add the following modules (CRLF at the end of each module and also end on a blank line): RESVC, REAPI, Routing, SMTP, AQ, CAT, EXSINK, StoreDrv.



Run the **Regtrace.exe** program to set the diagnostics (which also sets the other registry keys)



On the Traces page, select all the tracing types.

On the Threading page, uncheck “Write Traces on Background Thread”

On the Output page, increase the Max File Trace Size to 25 or 50MB. Toggle between “No Tracing” (turns tracing off) and “File” (enables logging to file). As soon as the test is finished, remember to turn tracing off.

The output is written to a file that can be analyzed with **Tracevwr.exe**

Message Tracking

If a replication message is not being delivered, message tracking is a useful way to find out where the message went. Once that has been determined, additional diagnostics can be set on the servers in the path to troubleshoot the problem.

Replication Problems

Permissions

Permissions are the most problematic area of Exchange 5.5 and Exchange 2000 co-existence. Be sure to determine whether the reason a client cannot see a folder is due to replication (the server has no knowledge of the folder) or permissions (the ACL has not been successfully upgraded to an NT SID.)

The best way to do this is to view the folder tree with ESM. If ESM can see the hierarchy, but a client on the same server cannot, then this is a permissions problem – NOT a replication problem.

Mixed mode Permissions Problems

Permissions problems fall into 3 categories.

1. Exchange 5.5 users ACL'd on folders are not in the W2K AD.

The store will log a 9551 event every time the folder is accessed. Exchange 2000 users will not be able to even see the folder until the problem is resolved. The 9551 event will indicate the user DN that is causing the problem. Either remove the user from the ACL or replicate the user to the W2K AD.

2. Object in W2K AD does not have a master account SID.

The store will log a 9548 event indicating which user is causing the problem. Again, no Exchange 2000 users will be able to even see the folder. Either remove the user from the ACL, or give them a Master Account SID (Associated External Account).

3. UDGs have not converted to USGs

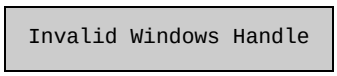
Most likely this is due to the UDG being in a mixed mode domain. Users who are members of the UDG will not be able to access the folder (unless they are explicitly ACL'd on the folder as well). Other users should be able to access the folder.

Losing MAPI permissions

On folders in the MAPI TLH you cannot mix the tools used to set permissions. MAPI aware tools such as ESM or Outlook should set MAPI TLH permissions.

If you set the permissions via Explorer or by ESM when viewing the NTSD permissions (CTRL → Client Permissions), you will break the MAPI permissions on the folder. The permissions can no longer be modified via MAPI.

Clients will get the following error if they try to modify the permissions:



More Information

Although Exchange 2000 allows you to set security on public folders in the public folder hierarchy and using Exchange System Manager, Outlook, and the Windows 2000 version of Windows Explorer, the tools are not interchangeable. This is because Windows Explorer uses the Windows 2000 access control list (ACL) format to set security permissions on the MAPI public folder hierarchy, and Exchange System Manager and Outlook use the MAPI ACL format. Exchange Web Storage System can correctly interpret both ACL formats, but the tools are not interchangeable. For this reason, you should only use Exchange System Manager when editing security on the MAPI public folder hierarchy. This problem does not exist on general purpose, or application folder, hierarchies. For example, if you originally use Windows Explorer to set permissions on a public folder, and then try to use Outlook or Exchange System Manager to change the settings, you will not be able to change public folder security until you follow the work around steps provided below. Then you should only use Exchange System Manager to set ACLs on public folders.

If the folder in question is a subfolder of **Public Folders** (Public Folders\TopLevelFolder), complete the following steps so that Exchange System Manager can be used to modify permissions.

To allow ACLs to be set in Exchange System Manager:

In Windows Explorer, right-click the appropriate folder, and then select Properties
On the **Security** tab, in **Name**, select an account, and then click **Remove**. Repeat this step for all accounts.
Click to clear the **Allow inheritable permissions from parent to propagate to this object**, and then click **Remove** on the confirmation dialog.
To save the changes, click **OK**
In Windows Explorer, right-click the folder again, and then click **Properties**
On the **Security** tab, select the **Allow inheritable permissions from parent to propagate to this object** check box.
To save the changes, click **OK**

If the folder in question is a 2nd level folder of **Public Folders** (Public Folders\TopLevelFolder\SecondLevelFolder), complete the following steps so that Exchange System Manager can be used to modify permissions.
To allow ACLs to be set in Exchange System Manager.

Complete the steps above for the TopLevelFolder.
Complete perform the steps above for the SecondLevelFolder.

Transports

Replication Messages not being received

Stores do not have email addresses.

Check that the RUS has correctly stamped the mail attributes onto the public store's directory objects.

In mixed Exchange 5.5 / Exchange 2000 organization, check that Exchange 5.5 can see the directory entries for the Exchange 2000 public stores and vice versa.

No route for mail to flow.

Check that normal mail traffic can flow between the servers.

If the replication message goes over an IMC, check that ResolveP2 is set and that the Exchange 5.5 DS object has been added as an X500 proxyAddresses in the W2K Active Directory.

Transport links restricted to disallow system messages.

Check that there is a route for system messages between the servers (**Winroute.exe** will show if there are restrictions on the links).

Replication

Backfill takes a long time.

This can happen when a new server is installed and the initial Status Request gets lost or goes to a server that also has no knowledge of the hierarchy. Make a change to the hierarchy on another server and check that it replicates through correctly. The server should backfill within 24 to 48 hours.

Emailing Folders

Mail message NDRs

Check the PF CA has replicated the folders directory objects correctly.

Remember that you cannot email App TLH folders from Exchange 2000, where the email message travels via an Exchange 5.5 server.

An email to a folder will need to go to a TLH server first, to find the replica list for the folder. It may be that the TLH server chosen has not received details of the folder yet.

Other

Cannot access a store via OWA, after the TLH has been renamed.

When you rename a TLH you have to update all the virtual roots that point to this TLH. Also the changes will not be picked up until after the database has been remounted.

Therefore, if you rename a TLH, you need to:

- Update the virtual roots on the servers that hold a copy of the TLH, so they point to the new one.
- Remount all the stores in the TLH so the changes get picked up.

Failure to do this will mean that OWA cannot access stores in the renamed TLH.

Error "Operation Failed" attempting to access a TLH via ESM

ESM uses an OLEDB layer called Rosebud to access the public folder trees. This relies on the WWW Publishing Service (W3SVC).

- Check that W3SVC is running on the Exchange 2000 server.
- Check that the Internet Explorer settings do no have a non-existent proxy server configured.

Exchange 5.5 servers see multiple Public Stores on an Exchange 2000 server.

This problem can occur if servers running an SRS are incorrectly removed from the organization.

This is a real problem for Exchange 5.5. This can occur if responsibility for writing Exchange 2000 MAPI public stores, from pure Exchange 2000 Admin Groups, into the Exchange 5.5 DS changes from one Config CA to another.

The new Config CA will not "see" that the Exchange 2000 MAPI store's object already exists in the Exchange 5.5 directory, because the object will have the old Config CA's Replication Signature.

It will re-replicate a duplicate set of objects, including the MAPI Public Store. This will cause a second MAPI public store to appear in the Exchange 5.5 Directory for the Exchange 2000 server. However, this store will have a DN of:

```
/o=<org>/ou=<pure Exchange 2000
site>/cn=Configuration/cn=Servers/cn=<Exchange 2000
server>/cn=Microsoft Public MDB - 1
```

This will cause the replication engines on Exchange 5.5 servers to fail to start ***throughout the organization***.

The following errors will be logged:

```
Event 3044 MSExchangeIS Public
Error 0x3f0 occured while performing a site folder teardown check

Event 3079 MSExchangeIS Public
Unexpected replication thread error 0x3f0
EcGetReplMsg
EcReplStartup
FReplAgent
```

Further Information

When Exchange 5.5 servers start, they perform a site folder teardown check, to see if any sites have been removed, in which case the list of site folders (e.g. Free & Busy etc.) needs to be cleaned up. This is done by comparing details about all the site folders with details about all the public stores in the organization.

As the string "Microsoft Public MDB – 1" is too long, the replication thread will error out with an Out Of Memory error (0x3f0) when it tries to get site details of this store. This will cause the replication engine to fail to start.

The only way to fix this is to remove both the incorrect directory object and the correct directory object for the Exchange 2000 public store from the Exchange 5.5 directory, and re-replicate the directory entry back in.

PSS must be contacted before attempting this, to ensure it is done correctly.

Useful Tips

These are some useful tips found when troubleshooting problems with Public Folders.

Before upgrading or Installing Exchange 2000

Make sure User CAs are replicating correctly before installing Exchange 2000.

If possible, the ADC should replicate Mailboxes and DLs to Native Mode Windows 2000 domains.

Run DS/IS adjust and remove unknown user accounts from the permissions lists of folders on Exchange 5.5. Be careful not to rehome folders.

Replication

If you think there is a problem with folder replication (especially hierarchy), use ESM to check whether folders have replicated. Do not rely on a client's view to determine whether folders have replicated. It might be a permissions problem, not a replication issue.

For replication issues, set diagnostics for Replication Incoming, Outgoing and NDRs to maximum.

If replication messages are not being sent / received, check that user's email between the servers works.

If a server doesn't appear to be backfilling, check whether new folders added to other servers replicate as part of hierarchy replication to the store. If they do then the server will realize it's not synchronized and write an entry into the backfill array (this could take 2 or 3 days to complete backfilling).

Mailing

You cannot email folders in App TLHs if the email will go via an Exchange 5.5 server.

Permissions

For Permissions issues, set diagnostics for Access Control, and Logon to maximum.

To view NT SIDs on folders, hold down CTRL when clicking client permissions in ESM.

Do not set permissions with explorer, or set permissions using the raw ESM interface (CTRL → Client permissions), otherwise you will never be able to set the MAPI-like permissions again.

Do not install Exchange 5.5 on a W2K DC. This is because the Exchange 5.5 directory will mask the W2K DC's RPC NSPI interface, which will prevent ESM and Outlook 2000 clients from being able to contact the directory to look up SIDs.

ADC

If the location of Exchange 5.5 public folders' directory entries is changed, you need to allow the Config CA to replicate this change through to the Active Directory and then restart all the Exchange 2000 public stores in the same site, so they pick up the new container. Otherwise Exchange 2000 will not change how it is building the LegacyExchangeDN attribute when a folder is created. (It is not expected that the Exchange 5.5 setting is changed often, normally if this value is changed it will be done immediately after Exchange 5.5 is installed and then left alone.)

Referral

If a folder only exists on Exchange 5.5, OWA clients cannot access it.

By default Exchange 5.5 connectors are set to Disallow Public Folder Referrals.

General

Clusters can only contain one database in for each TLH stored on the cluster. For example you can only have 1 MAPI Public Store (pubx.edb) in the cluster. This is because if the cluster is in active-active node and it fails over, you would otherwise have multiple MAPI stores controlled by the same node.

When removing public stores, always replicate the folders to at least one other server, or you will lose the content.

When removing Exchange 2000 servers, always run setup → remove all, do not simply remove the server from the Organization.