# SHARING

When a share is created there are 3 principal components:

1. The Share creator (User\_id -> Account\_id)
2. The Share recipient(s) (email\_address = Share\_id)
3. The object(s) being shared (Share\_Folder or Shared files)

Steps to create a share

Every person shared with is an Account User. They are either User on a different Account OR they are Users on the Guest Account (Account\_id = 0). If they are neither, then they are added as Users of the Guest Account. An appropriate Folder structure has to exist to put the shares into and to be able to control them.

**STEP 1)** Record from the Interface the common information (account\_id, user\_id, date, message) into the Share table. The next step captures the details of the shares into the Shares\_Items table. This is a classic Order -> Details schema.



**STEP 2)** is processed in a loop against the Share recipient: (For i as int = 0 to Recipients.count -1) or (For each recipient in Recipients etc.).

Determine if the recipient is a User; (SELECT Account\_id, user\_id, email\_address FROM Users WHERE email\_address = @recipient). If the count of returned rows is 1 then we have an existing user. Store the Account\_id and the User\_id.

If the account is 0 then we need to add the recipient as a user to the Guest Account:

INSERT INTO Users (Account\_id, email\_address, password) VALUES (0, recipient’s email address, ‘1QAZ!qaz’). Refer to the @@IDENTITY as @NewUser

Store the @@IDENTITY of the new User. In the same Stored Procedure, insert a Guest Root Folder of Folder\_type = -3 to the Folders table:

INSERT INTO Folders (Folder\_Type, Folder\_name, Account\_id, User\_id) VALUES (-3, recipient’s email address, 0, @NewUser). Refer to this as @GuestRootFolder

Update the User record with the value of the newly created Guest Root folder:

UPDATE USER SET [Guest\_Folder\_id] = @GuestRootFolder.

For a *new* Guest Account User you must add a Host Account and Host User folder. For an existing User (either Guest or Standard) you *may* have to add a Host Account and Host User folder.

Adding Host Account and Host User Folders

New Guest User: We know they do not exist. For an existing user check 1st for existence.

INSERT INTO FOLDERS (Folder\_type\_id, Folder\_parent\_id, Folder\_name, Account\_id, Host\_Account\_id, User\_id) VALUES (3, @GuestRootFolder, (Name of logged on Account), 0, (Id of logged on account), @NewUser). Refer to the Id of this new folder as @HostAccountFolder.

INSERT INTO FOLDERS (Folder\_type\_id, Folder\_parent\_id, Folder\_name, Account\_id, Host\_Account\_id, Host\_User\_id, User\_id) VALUES (4, @HostAccountFolder, (Name of logged on User), 0, (Id of logged on account), (id of logged on user), Id of Share). Refer to the Id of this new folder as @HostUserFolder.

Now the necessary Parent Folders for the new share are in place. New shares of 1 or more files or folders each are contained in a Share folder which can records the master id of the Share! In this manner, the Share can keep track of when it has been accessed. The folder name is the date/time of the share creation. The parent of this new Folder is @HostUserFolder.

INSERT INTO Folders (Folder\_type\_id, Folder\_parent\_id, Folder\_name, Account\_id, Host\_Account\_id, Host\_User\_id, User\_id, Share\_id) VALUES (5, @HostUserFolder, “Month/DD/YYYY HH:MM AMPM”, 0, (Id of logged on account), (Id of logged on user), (Id of Share). Refer to this as @ShareFolder.

For each File in the share – add the ACL from the rights created on the share.

INSERT INTO Folders\_Files (Folder\_id, File\_id) VALUES (@ShareFolder, Id of File)

INSERT INTO Folder\_ACL (Folder\_Id, User\_id, Read\_only, Write, Share, Download, Upload, Printer, ACL\_create\_user\_id) VALUES (@Share Folder, Share\_id, @Read\_only, @write, 0 , @Download, @Upload, @Printer, (id of logged on user)

If the item being shared is a Folder, one additional step is required: make a virtual copy of the folder being shared. It is a copy of the existing row with minor changes. An ACL entry is also made

INSERT INTO Folders (Folder\_type\_id, Folder\_parent\_id, Folder\_name, Account\_id, Host\_Account\_id, Host\_User\_id, User\_id) VALUES (5, @ShareFolder, (Name of Folder being copied), 0, (Id of logged on account), (Id of logged on user), (Id of Share). Refer to this as @ShareFolder\_Sub.

Virtually copy the original Files to the new folder

INSERT INTO Folders\_Files (Folder\_id, File\_id) VALUES (@ShareFolder\_Sub, Id of File)

INSERT INTO Folder\_ACL (Folder\_Id, User\_id, Read\_only, Write, Share, Download, Upload, Printer, ACL\_create\_user\_id) VALUES (@ShareFolder\_Sub, Share\_id, @Read\_only, @write, 0 , @Download, @Upload, @Printer, (id of logged on user)

**STEP 3)** The last step is to update the Share table with the location of the @ShareFolder. In this manner, the Account that created the share can manage the existence and rights of the share.

UPDATE SHARES SET [Share\_Folder\_id] = @ShareFolder

This establishes a link between the creator of the share and the recipient of the share. Access of shared files can be recorded and the share owner notified.

GUEST ACCOUNT VIEW OF FILE CABINET

The Guest user sees the Filing Cabinet with the same layout as the Account User. However, there is only 1 work space (No company space, personal space etc.). The tree view of folders is on the left panel. It is grouped by 1) Host Accounts who have shared items – more than 1 Filocity account may share items with a Guest user. Below each Host Account is a folder for each Host User of that Host who has shared items. Below each Host User are the folders for the different shares he has created.

GUESTS ROOT FOLDER: Folder\_Type -4 (not visible)

* Host Account Folder: Folder\_type 4
  + Host User Folder: Folder\_type 4
    - Share Folder: Folder\_type 5
    - Share Folder: Folder\_type 5
  + Host User Folder: Folder\_type 5
    - Share Folder: Folder\_type 5
* Host Account Folder: Folder\_type 3
  + Host User Folder: Folder\_type 4
    - Share Folder: Folder\_type 5

The Type of Folder for each level triggers the main section of the cabinet.

# WHEN A SHARE UPLOADS

Assume a share has the right to upload files. They can put them into any of the existing folders they have.

1. The file is recorded in the file table
2. An entry is made in the Guest accounts target folder
3. An entry is made in the Host

### TYPES OF FOLDERS

### Example SP to enter a guest user and retrieve new ID and new Host\_User\_Folder\_Id.

USE [Filocity\_2012]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[usp\_Guest\_User\_Insert] Script Date: 04/30/2012 08:34:24 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: LMW

-- Create date: 4/29/2012

-- Description: Inser a user into the Guest Account

-- =============================================

CREATE PROCEDURE [dbo].[usp\_Guest\_User\_Insert]

(

@Account\_id int = NULL

,@User\_id int = NULL

,@Email\_address varchar(100) = NULL

,@Password varchar(32) = '1qaz!QAZ'

,@New\_user\_id int = NULL OUTPUT

,@New\_Host\_user\_id int = NULL OUTPUT

)

AS

BEGIN

SET NOCOUNT ON;

-- Declare holding variables for daisy chaining folder entry

DECLARE @New\_Root\_id int

DECLARE @New\_Host\_Account\_id int

INSERT INTO Users(Account\_id,Email\_address,User\_password)

VALUES (0,@Email\_address,@Password)

SET @New\_user\_id = @@IDENTITY

-- Insert a Root Guest Folder

INSERT INTO Folders (Folder\_type\_id,Folder\_name, Folder\_owner,Account\_id, Host\_Account\_id,

Host\_User\_id)

VALUES(-3,'Guest Root',@New\_user\_id,0,@Account\_id,@User\_id)

SET @New\_Root\_id = @@IDENTITY

-- Update user record with guest root folder information

UPDATE Users SET Guest\_folder\_id = @New\_Root\_id WHERE User\_id = @New\_user\_id

/\* Insert a Host Folder for the guest; the only reason that a guest is created is for a share \*/

INSERT INTO Folders (Folder\_type\_id, Folder\_parent\_id,Folder\_name,

Folder\_owner, Account\_id, Host\_Account\_id, Host\_User\_id)

VALUES(3,@New\_Root\_id,(SELECT Account\_name FROM Accounts WHERE Account\_id = @Account\_id),

@New\_user\_id,0,@Account\_id,@User\_id)

SET @New\_Host\_account\_id = @@IDENTITY

/\* Insert a Host User folder for the guest; the only reason that a guest is created is for a share \*/

INSERT INTO Folders (Folder\_type\_id, Folder\_parent\_id,Folder\_name,

Folder\_owner, Account\_id, Host\_Account\_id, Host\_User\_id)

VALUES(4,@New\_Root\_id,(SELECT [First\_name] + ' ' + [Last\_name]FROM Users WHERE [User\_id] = @User\_id),

@New\_user\_id,0,@Account\_id,@User\_id)

SET @New\_Host\_user\_id = @@IDENTITY

SELECT @New\_user\_id, @New\_Host\_user\_id

END

GO

The Procedure can be re-written to encapsulate the process into a Transaction. It could also be changed to handle a new guest or an existing guest who would already have a guest root folder. Then it would check for existing Host\_account\_folder and existing Host\_user\_folder.

### FOLDERS, FOLDER\_TYPES, FILES

