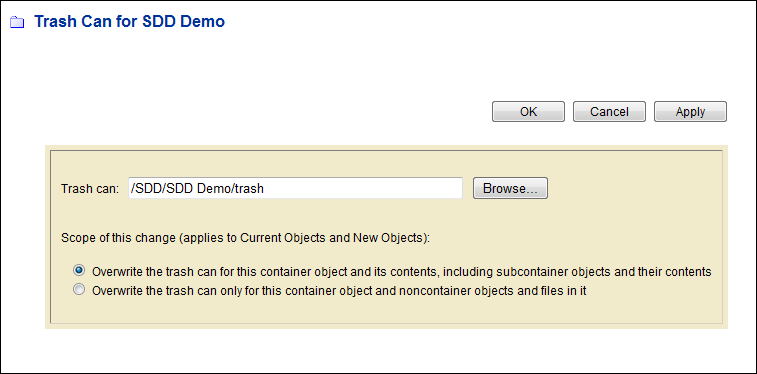
**Trash Cans: General Information**

1. A Trash Can can be defined for any container object in SAS Drug Development. It’s an intermediate holding area for objects that are deleted in a folder. A user can move an object out of a trash can using cut/paste, or perform any other action on that object. To completely delete an object, it needs to be deleted from the trash can.
2. There are two options when defining a trash can. A Trash Can can be set up to collect all deleted objects in the container object as well as all subcontainers. Or it can just be set up for the current container object.

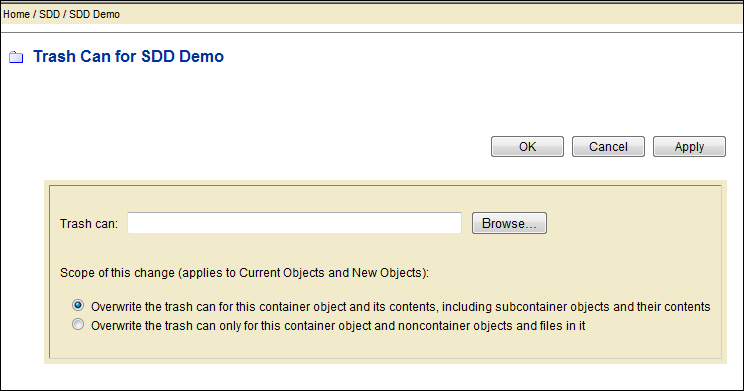
Note: when it is set up for all subcontainers, that will also include any new folder that is created anywhere in the hierarchy below the trash can, even if that folder is created after the trash can is defined.



1. A Trash Can does not appear in a folder until an object is actually deleted.
2. If an object is sent to the trash can containing an object with the same name, the object being deleted will now be named with a date/time stamp to distinguish it from first object. (Example if mydoc.docx already exists in the trash can, the newly deleted file will have the name: **mydoc.docx20100812190535)**
3. If a versioned object is sent to the trash can, the versions are maintained in the trash can.
4. A User must be assigned the **User Can Manage Trash Can** policy in order to **create** or **remove** trash cans.

**Deleting Versus Removing a Trash Can**

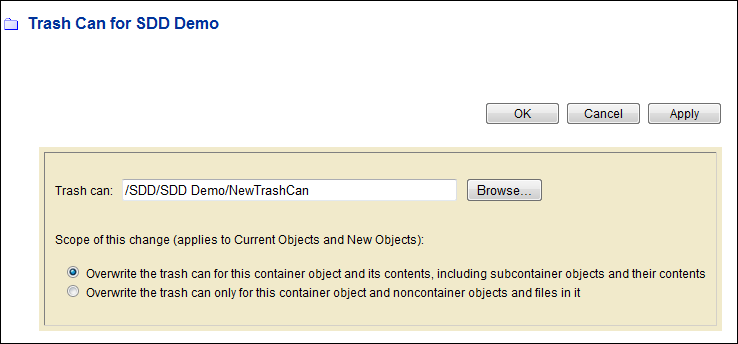
1. When a trash can is created, a pointer is then associated with that folder and the trash can. If the trash can is created for all subfolders, then each subfolder and any new subfolder will contain a pointer to that trash can.
2. If you delete the trash can from the directory, the pointer still remains. After the trash can is deleted, a new trash can will appear as soon as an object is deleted.
3. To **remove** a trash can, select Manage Trash can for a folder. The name of the trash must be “blanked out” in order for the pointer to be removed. This change in the trash can pointer can be applied to all subfolders (default) or just the current folder.



1. A User with “delete” access on the trash can delete it, as long as they have permission to delete all of the objects within the trash can. A user must be assigned the **User Can Manage Trash Can in order** toremove the trash cans.
2. You can remove or change the trash can pointer without deleting the existing trash can.

**Changing a Trash Can**

1. You can change the trash can for a folder, by defining a new name in the Manage Trash Can window. Any object deleted now will be directed to this new trash can. The old trash can will remain in the directory until it is deleted.



1. Creating, Changing, or Removing a Trash Can is recorded in the Audit Trail.
2. You can turn the trash can off or change it for a folder in a hierarchy, even if the folders above it and below it are directed to the trash can. You will need to go select “Manage Trash Can” for that folder and either blank out the name of the trash can to remove it or change to a new name. Select the option to “Overwrite the trash can only for this container object and noncontainer objects and files in it”. (2nd option).

**Things to know about the Trash Can**

1. **Limits on Number of Objects:** There are no limits to the number of objects that can be stored in a trash can. However there are limits to how many objects can be displayed in a container as well as the actions that can be done on these objects.
2. **Navigation:** When navigating folders with large numbers of objects, the SDD User Interface is better at displaying large number of objects than webfolders or the SAS Drug Development Desktop Connection (SDDDC).
3. **Errors with webdav :** When deleting large numbers of objects with the SDD Desktop Connection or a Web Folder, an error dialog may be presented which indicates the operation failed. Typically, this can be ignored and the user should allow enough time for the delete operation to complete and verify if the delete operation was successful.
4. **Deleting the trash can folder versus deleting large numbers of objects in it:** A user is less likely to have problems deleting the trash can folder than selecting all or many of the objects for deletion, when deleting very large numbers of objects.
5. **Large files:** If you’re trying to delete files from a trash can using SDD Desktop Connection or a web folder, the files that are 2 GB or larger will not be displayed in SDD Desktop Connection. You will need to view and delete these files through the SDD UI. If only using web folders or the SDD Desktop Connection, users may think that a trash can is empty, when actually it could still contain large files.
6. **Performance considerations:** Deleting objects permanently is quicker than using a trash can.
7. **Overwriting Files:** When files are created/overwritten in SDD (either from the Process Editor, Scheduled Job, Basic or Advanced Loader) and there is a trash can set up for the folder, the files will go to the trash can, but then will be immediately deleted from there. It works this way so that there are not items with duplicate ids in the trash can. It takes a lot longer to create or load data if the overwritten files are first sent to a trash can and then deleted. If there is not a trash can defined to the folder, the creating or loading of data is much more efficient.
8. **Audit Trail:** If an item is deleted and sent to the trash can, the audit trail will indicate that it’s been deleted, but there is not information that it’s been sent to the trash can. If it’s deleted from the trash can, there will be another record in the audit trail of the deletion. The records in the audit trail indicate the location of the file prior to deletion. So in the case of an object deleted from the trash, you’ll see the path of the trash can in the audit trail record.
9. **Original Path Metadata:** The original path of the file is not part of the metadata for an object in the trash can. However this information is stored in the SAS Drug Development Database and, if needed, can be requested to be extracted with a query.
10. **Modified Date Metadata:** The modified date is not updated when on objected is deleted and sent to the trash can. Therefore it’s not possible to see how long objects have been in the trash can. It’s only possible to see when the object was last modified prior to deletion.
11. **Remote api:** Currently the remote api does not support setting trashcan attributes.

**Best Practices for Setting up Trash Cans**

1. **Always choose the 2nd option in the Manage Trash Can window**

“Overwrite the trash can only for this container object and noncontainer objects and files in it.”

In this way, you will know exactly the folders that are being directed to a trash can.

The default behavior is for the trash can to be set up for the current folder and all subfolders below it in the hierarchy. If a user creates a trash can on a higher level or a top-level folder using this default, then all folders below it will point to that trash can as well as any new folder that’s created anywhere in the hierarchy below it. This can result in lots of files being sent to the trash that don’t necessarily need to be saved. This can also result in trash cans (defined on lower level folders) being sent to other trash cans (defined on higher level folders) when they’re deleted.

Do not put the trash can on a folder in a top-level directory and with all of the files in the subfolders below it going into the trash can.

2. **Protect important data and programs by setting restrictive permissions and versioning.**

A Trash Can could be set for folders with important data, but it's unlikely that it would fill up since permissions would prevent deletions. Programmers may want to use checkin/checkout on their SAS programs. When using checkin/checkout, a new version of the program is only created when it's checked in, so new versions are not created with each modification.

**3. Do not set up trash cans on folders containing overwritten files**.

For files that are overwritten (through a data load or SAS program), it is probably not necessary to have a trash can, since the source data is in another area. It’s best to remove the trash can from those directories since there are performance implications and the overwritten files are not saved in the trash can anyway.

1. **Remove trash can folders from user sandbox areas, testing and training areas.**

Large numbers of files area created and deleted in these areas and are typically not essential. Users can always protect important files by permissions and versioning.

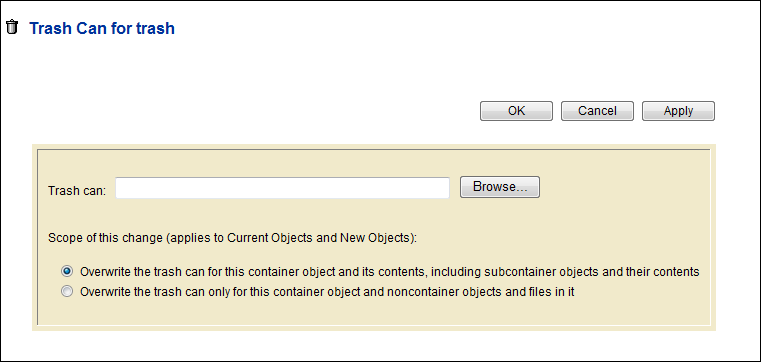
1. **Define a process for deleting files from the trash can on a regular basis.**

Objects can be deleted using the SDD User interface. Web folders and the SDD Desktop Connection can also be used, but objects larger than 2GB will not be displayed. A SAS process can also be created which uses the api macros to delete files from a trash can. Specific criteria can be defined for deletion, such as only deleting objects that were modified more than 20 days ago.

It would be best to schedule this program each evening so that the trash can is cleaned up on a regular basis. Note: The SDD implementation group can provide an example program that uses the api macros to delete files from the trash can. This program also can calculate how many days an object has been in the trash can and delete based on that criteria.

1. **Do not define a trash pointer for a trash can container.**

The trash can pointer should be blank for trash cans. If you have a trash can pointer on a trash can that points to itself, it will not be possible to delete anything from the trash can and error messages will be generated. It’s best not to define any trash can for the items that are already within a trash can.



[ Show » ]

[Mark Matthews](https://www.sso.sas.com/jira/secure/ViewProfile.jspa?name=sasmdm) added a comment - 01/Jul/10 12:46 PM Mark, Following our discussion yesterday, here is what we did with the system and might be responsable for the instability of the system. 1. At the root, there is a folder GSKBIO / PROJECTS. It contains all the data related to our clinical studies, grouped by clinical project. Example: /GSKBIO/PROJECTS/Project A/STUDIES/Study 1/... Today, there are 47 projects, probably 200+ studies and the total size of the PROJECTS folder is about 90 Gb. 2. At the setup of the environment, I activated the trash can on the PROJECTS folder thinking that it would backup all files and folders that might be deleted. I now realize that it does keep all versions of the deleted files but in a unique folder. In other words, and correct me if I did something wrong, it does not keep the location of the deleted files. So if I have two files named CHKWIN.SAS in two different studies and I delete both files, the two files are saved in the trash can (/GSKBIO/TRASH) but there is no way to identify which one was in study 1 and which one was in study 2... Correct? 3. We are planning to add an additional level between the project and the study. In the new structure, studies will be in /GSKBIO/PROJECTS/CLINICAL/Project A/STUDIES/Study 1. 4. As a test before the migration, a few weeks ago, I tried to copy all the projects in the new CLINICAL folder. Note that to make the copy, I used the web interface. I selected all the projects, Actions -> Copy. I went to the CLINICAL folder, Actions -> Paste. The copy took a (very) long time but when I came back to the office the day after, it was done. We found some inconsistencies in the size of some folders when comparing the PROJECTS folder and the CLINICAL folder but the content seems to be ok. 5. Yesterday, we decided to delete the whole content of the CLINICAL folder (probably more than 40 Gb) and that's when the problems started. The user who tried to delete the folders was kicked out of his session several times. Seeing that the folders were still present, we tried to delete the folders again (it was probably still running the first process though). We even tried to delete the folders via the Desktop Connection (probably a bad idea). 6. Thinking that the trash can might be responsible, I tried to remove the trash can on the Projects folders. I was also kicked out of my session, several times. 7. At that time the system was down, impossible to connect via the web interface nor desktop connection. Servers were rebooted. 8. Things are back to normal now but I have a trash can of about 24 Gb that is totally unexploitable. Actually, it's impossible to browse its content. I tried to delete it when it does not work. My concerns are: - We currently have no backup in case a user comes to me to ask the restore of a file or a folder. Do you have any advice on the usage of the trash can? - I don't feel very comfortable with a CUT/PASTE of the whole projects to move them in the new hierarchy. That's the reason why I first want to make a copy. Considering the time it took a few weeks ago and the unstability when we deleted a large number of files, I feel nervous about the migration on July 16th. Is there any other reliable way to move files within the system? Or to monitor the process as it is running? - I have a 24 Gb trash can but there is nothing I can do with it, not even delete it. Thanks, François

**Workarounds for Problem Areas**

1. **Starting over with trash cans:**

If you already have trash cans set up and you need to start over, you can do the following**:**

* **Manage Trash Can on a top-level folder.**
* **Set the trash can to “blank”.**
* **Choose the default option of overwriting all folders below it.**

This will remove the trash can pointer from any folder in a below a top-level folder in a hierarchy.

1. **Managing a trash can that is too big:**

You can change the trash can pointer so that a new trash can is used for that folder. The old trash will still remain in the folder until it is cleared out. It can be cleared out through the SDD UI or through a process. Or, you can wait until it’s likely that nothing will need to be recovered and then the entire folder can be deleted.

1. **Unable to delete an object or the entire trash can:**

The Xythos Admin tool can be used to delete files or a trash can. However files deleted with the Xythos Admin are not recorded in the audit trail.

1. **Unable to open trash can because of large number of objects:**

The Advanced Search function can be used to retrieve items from the trash can even when you’re not able to open it. Objects can be deleted or moved from the search results page.

1. **Trash can is growing, but you’re not ready to permanently delete the objects:**

You could “archive” the trash can. That would create a zip file in the archives folder that contains all of the objects in the trash can. You can then delete the trash can and start over with an empty one.