Filestore Form Overview

The Lattice Filestore app seeks to mimic the functionality of the network shares commonly used in organizations, combined with some of the functionality from OneDrive or DropBox, but is built as a distributed grid system in order to add quite a few useful features. The best way to explain those features is to walk through the functionality using some realistic examples.

Folders Tab Functionality

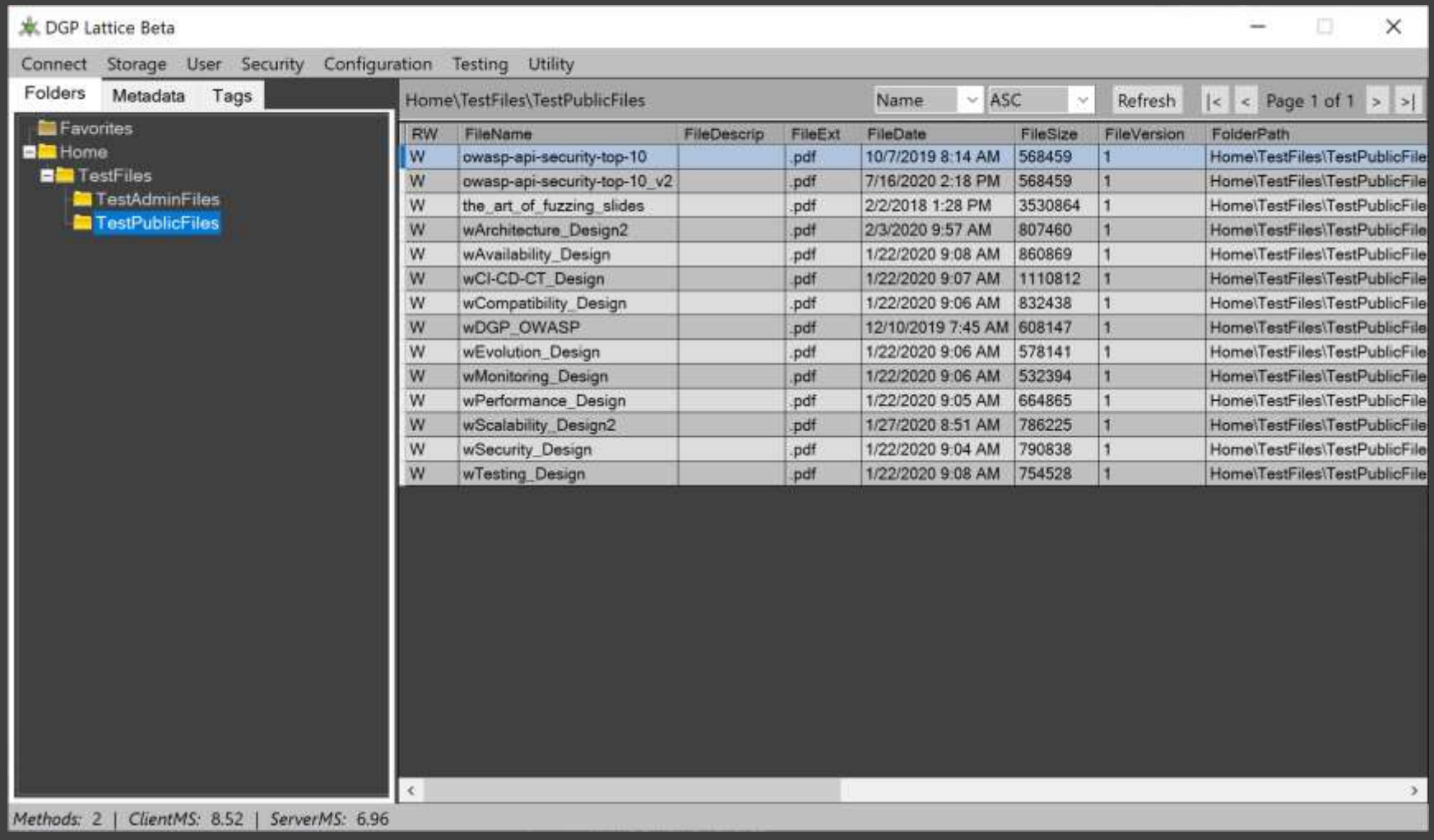
When people start using a network share, DropBox, etc. their first task is to create folders to store and organize the files. In Lattice, this is done using the Folders tab on the left side of the form… but users will immediately notice a few differences. An empty new Filestore system will show two folders already present: Favorites and Home. These are both system folders that cannot be deleted or changed.

The Favorites folder acts in much the same way as a list of favorites in a web browser. It is a list of frequently used files that each user maintains for themselves, and which should allow users to quickly return to frequently used files faster than drilling down into the folder tree or searching for a specific file every time. Files are not stored in the Favorites folder – it just contains links to files stored elsewhere in the directory structure. No right-click context menu is displayed for the Favorites folder, which means users cannot edit it or add subfolders to it.

The Home folder is basically the root of the virtual “network share”. All other folders will be created as subfolders under it. The right-click context menu only allows users to add subfolders to the Home folder, and nothing else. Users are not allowed to upload files to the Home folder or the Favorites folder.

When creating folders, users will notice that they must choose a DataGroup to which the folder belongs. The dialog form to create subfolders contains a list of the read-write DataGroups that the user belongs to. All users are allowed to create subfolders under the Home folder, as long as they have read-write access to at least one DataGroup (which they will – if they don’t that is an error in the setup of their user account to use the FileStore application). DataGroup membership acts as an access control mechanism to restrict which folders and files each user is allowed to see and use. The basic premise is that the users of the system should NOT be able to

see and/or edit all content stored in the system, but only the content they have been authorized to see and work with (similar to the authorization mechanism used for API methods).



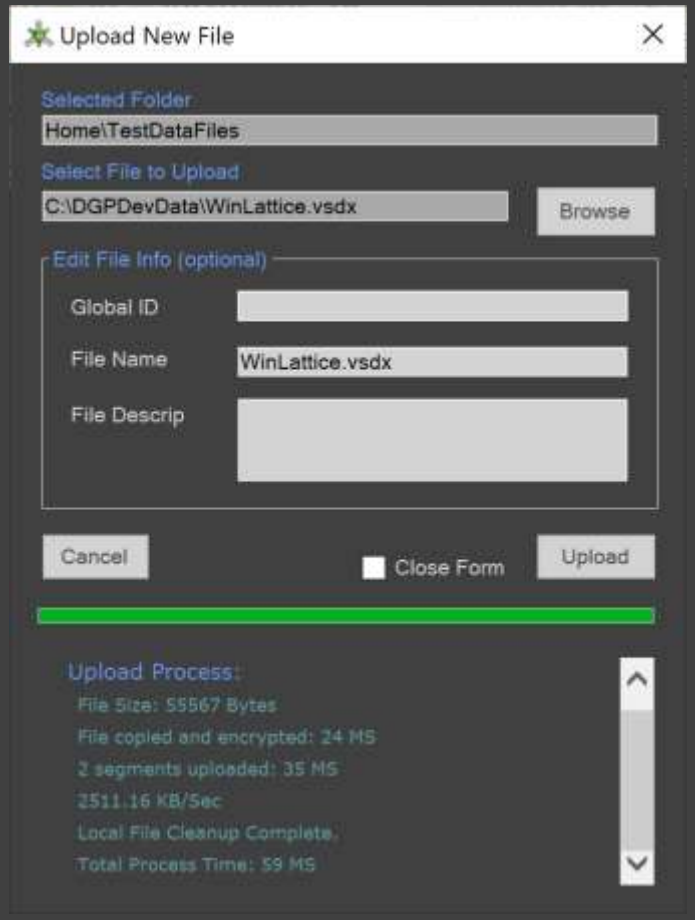
Gray folders in the directory tree are ReadOnly for that user account. Users cannot upload files to them, add subfolders, or edit/delete them. All users can do is download files from read-only folders. Gold folders in the directory tree are ReadWrite for that user account. Users can upload files to them, create subfolders, and edit/delete the folder itself. Folders can only be deleted when they contain no files or subfolders, so they have to be cleaned out first (to avoid creating "orphaned" files in the system).

Folder Context Menu	Description
Upload New File…	Displays the File Upload dialog, which allows the user to browse for a file to be uploaded to the selected folder.

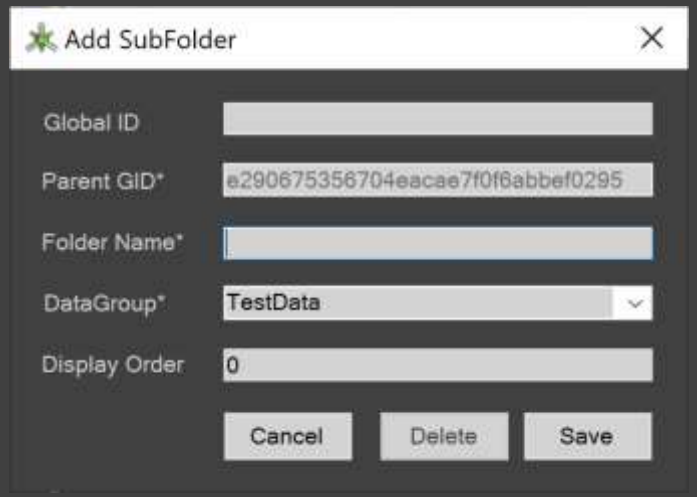
The first step of the file upload process is to collect information about the file for the File metadata record. The info is read from the file to be uploaded and reused from a previous File record, or created for a new File record (the GroupGID, FileGID, and FileVersion are provided by the system).

The next step divides the file into segments, encodes the segments as Base64 text, and uploads the segments one at a time to create one or more FileShard records linked to the File record as a parent. The file segments can be encrypted, but for most systems that extra work is not necessary. Due to the size of the file segments and the limitations on the size of API messages, the file segments cannot be batched – they must be processed one segment at a time, sequentially.

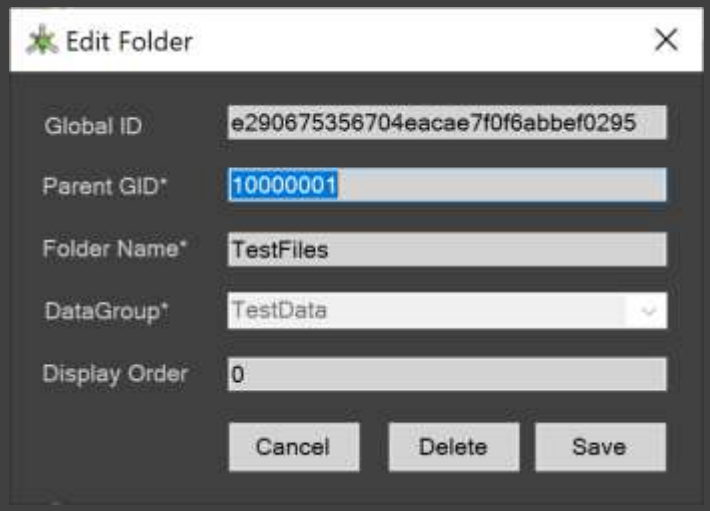
If the upload of the segments is successful, then the final step is to create a new parent File record or update the existing record. If either the file upload step or the parent record step has a problem, the system is returned to its previous state by deleting any FileShard records that were created prior to the problem.



Folder Context Menu	Description
Add New Subfolder…	Displays the Add Subfolder dialog which allows the user to add a new subfolder to any parent folder for which they have read-write access. When a user creates a subfolder, they must select which DataGroup "owns" the subfolder from the list of read-write DataGroups the user has access to. The Display Order value can move a folder up or down in a subfolder tree, overriding the default alphabetic sort order.



Folder Context Menu	Description
Edit Folder…	Displays the Edit Folder dialog which allows the user to change the Parent GID, Folder Name and Display Order values, or to delete the folder (folders that contain subfolders or files cannot be deleted until they are empty). Editing the Parent GID value moves a folder (and all of its subfolders) to a new location in the folder tree. Users can only move folders to a new parent folder for which they have ReadWrite access (enforced within the web service API methods).

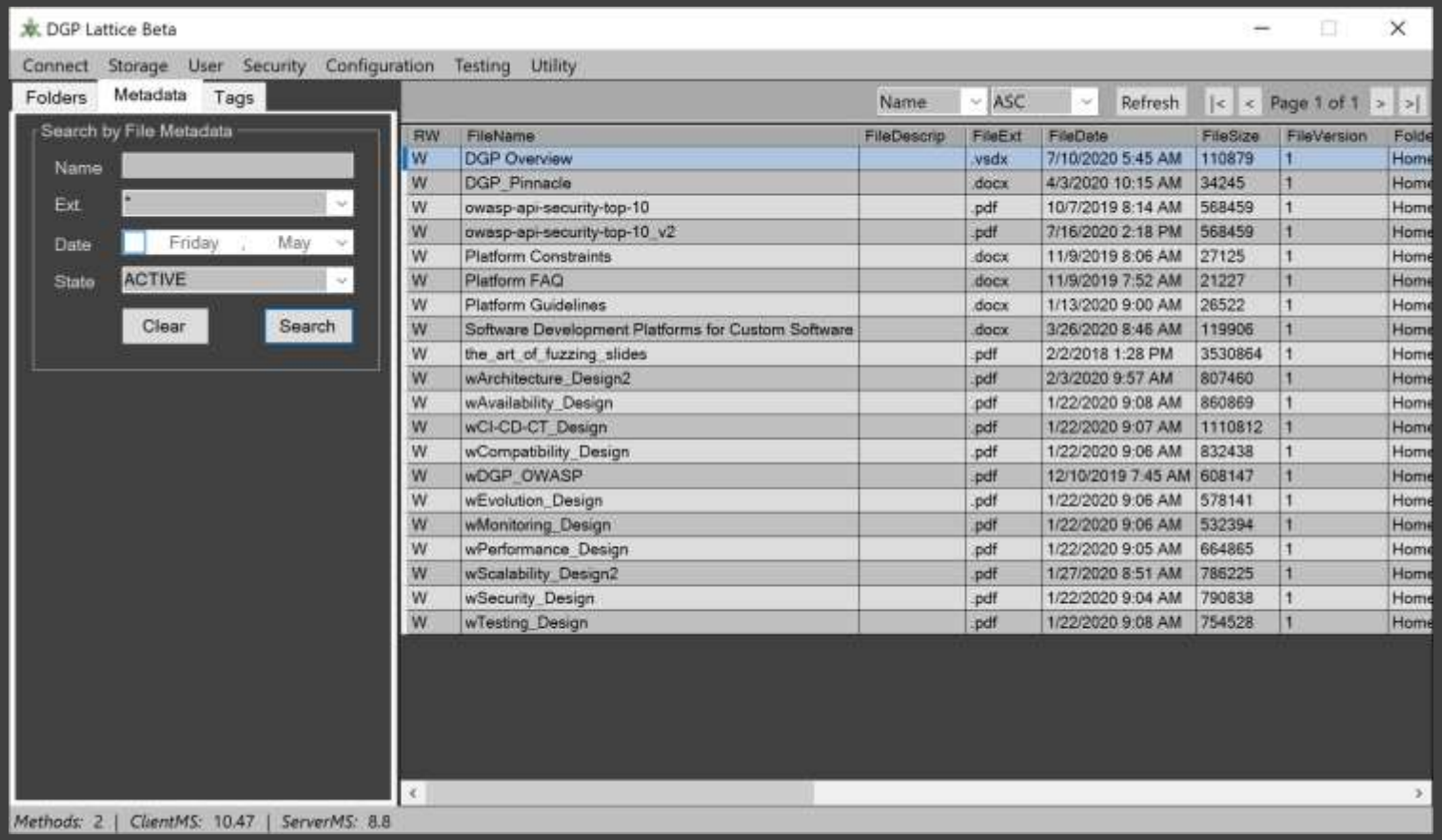


Due to the complexity of merge replication and the different access levels of folders, moving folders is currently a manual process (but could be done with Drag-and-Drop in the future). On the other hand, copying folders, all their subfolders, and creating new link records for all the files contained in each one could result in a large batch of work to be done on the server. Techniques to do this efficiently (without becoming a mechanism for DDOS attack) are currently being considered.

Metadata Tab Functionality

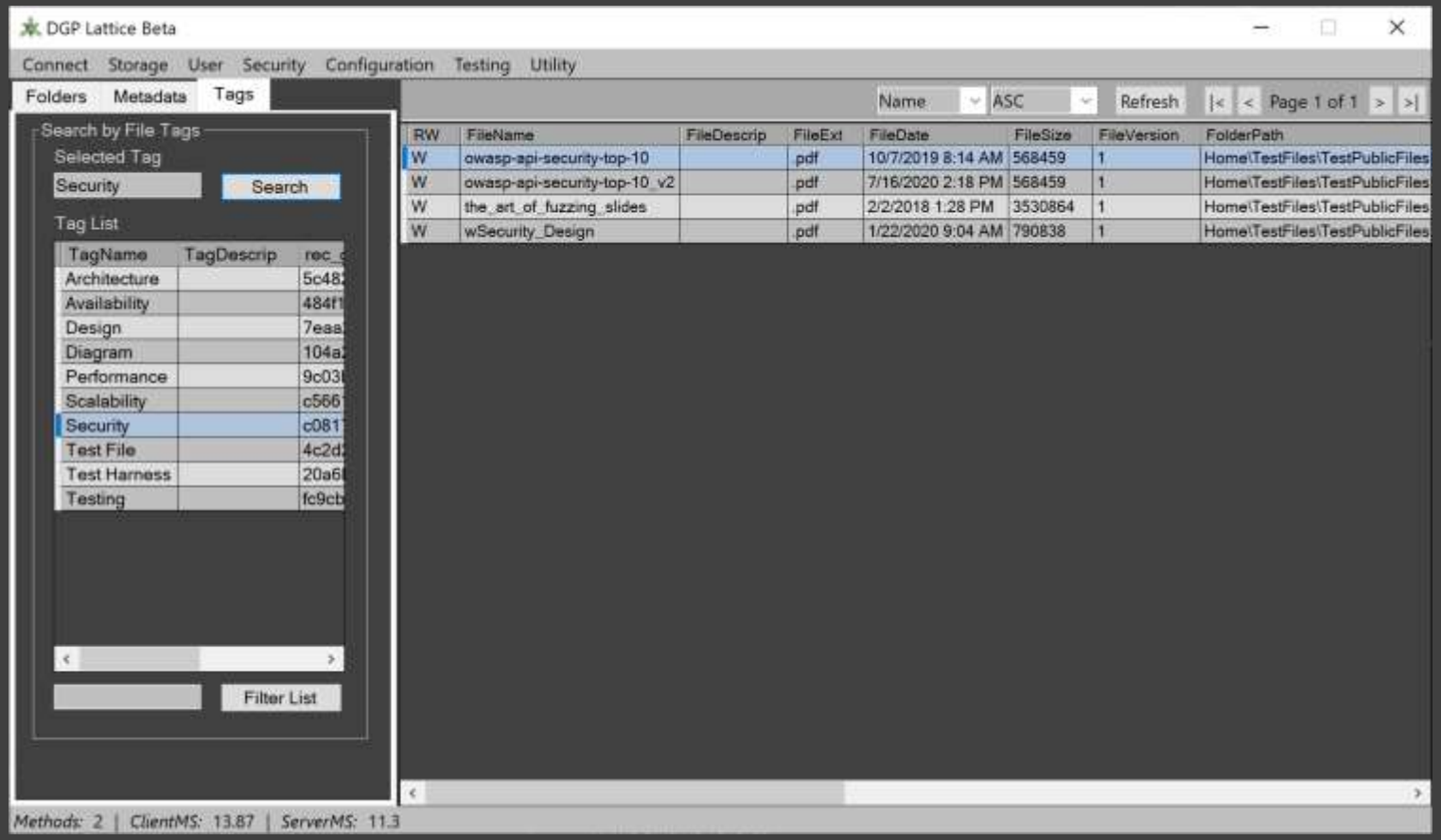
This section allows users to search for files by their file name, the file type (file extension), file date (newer than the selected date), and file state (Active or Deleted). More than one filter criteria can be used at once, progressively narrowing down the number of files found in the search results.

Note: searching for files with the Deleted state is how deleted files can be recovered.



Tags Tab Functionality

Double-clicking on a tag in the Tag List grid will populate the Selected Tag text box. Clicking the Search button next to it will populate the file grid with the first page of files linked to the selected tag. Tag search is currently limited to a single tag at a time due to the difficulty of assigning a list of values to an SQL parameter. The tag list can be filtered using the textbox and Filter List button below the Tag List grid.



File Info Grid

All of the different functionality on the left side of the form act as different mechanisms to populate the same grid of file info records on the right side of the form. When using the Folder tree to browse for files, the directory path of the selected folder will be displayed on the upper left area of the file grid.

The 3 combo boxes next to the directory path select which field to use for sorting the pagination query, whether the results are displayed in ascending or descending order, and the number of records in each page of results. These values are passed into the Search API method call, and affect the pagination of the results shown in the file grid. All other search filter values are optional, and clicking the Search button with no filters will simply return the first page of records in the Files table.

The Refresh button will run the search query again after having changed the selection of any of the combo boxes.

The next series of buttons to the right of the combo boxes control the user's navigation through the pages of file grid search results. All pagination is executed as SQL syntax on the server, so the pagination buttons simply change some of the values passed into the Search API method call.

All calls to Search methods automatically apply SQL syntax to filter search results based on the DataGroup membership of the user. Also, the first column in the file grid ("RW") indicates if the file record in the grid is read-only ("R") or read-write ("W") for that user account.

File Info Grid – Context Menu

The navigation items in the file info grid context menu vary based on which method was used to populate the search results, and whether the file info record is ReadWrite (W) or ReadOnly (R).

ReadOnly files show:

- Download File…

ReadWrite files show:

- Download File…
- File Tags…
- Upload New Version…
- Edit File Info…
- File History…

Favorites files include:

- Remove From Favorites

All other sources include:

- Add To Favorites

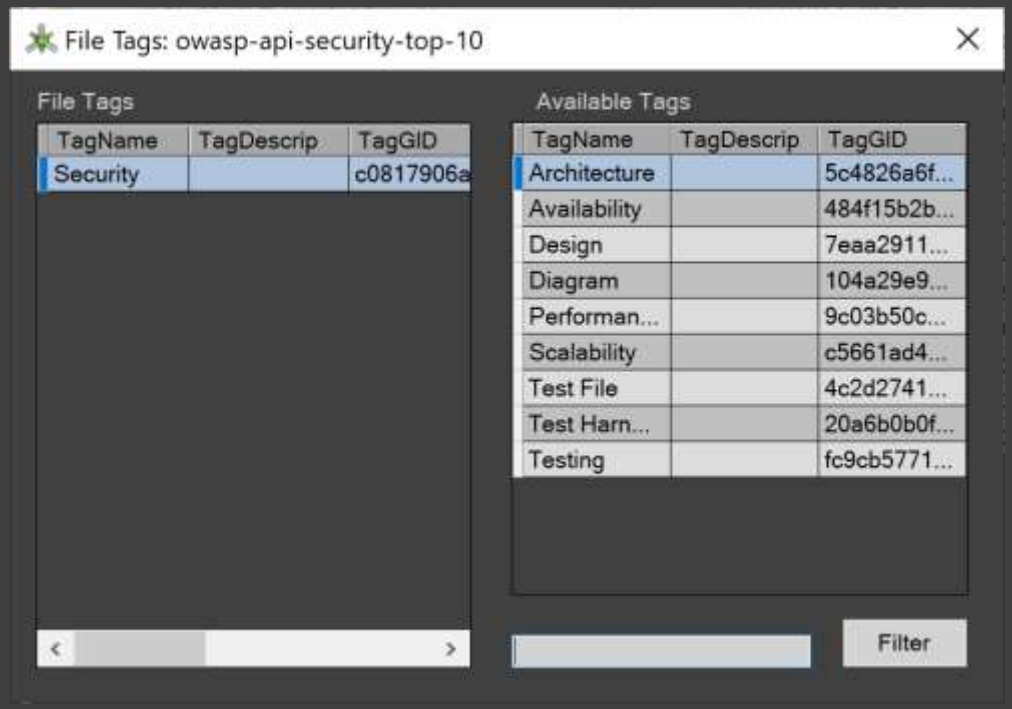
File Context Menu	Description
Download File…	Shows the File Download dialog populated with data from the selected file info record. The user selects a local directory in which to download the file, and whether or not to close the form and open the file automatically once it has been downloaded.

The File download process uses the information from the selected file to call an API method for a list of the FileShard records linked to the parent record. The records in the result do not contain the file segment data (they would be much too large as a collection). It then iterates through the list to call an API method to retrieve the data for each file segment in sequential order, convert it from Base64 back to its original format, and append the segment data to the file downloaded to the selected local directory. Once the file has been downloaded, a SHA256 hash is calculated and compared to the File record value. If they match, the downloaded file is named according to the file info record (or user supplied name), and the file date is set to match the file info file date. If the user has

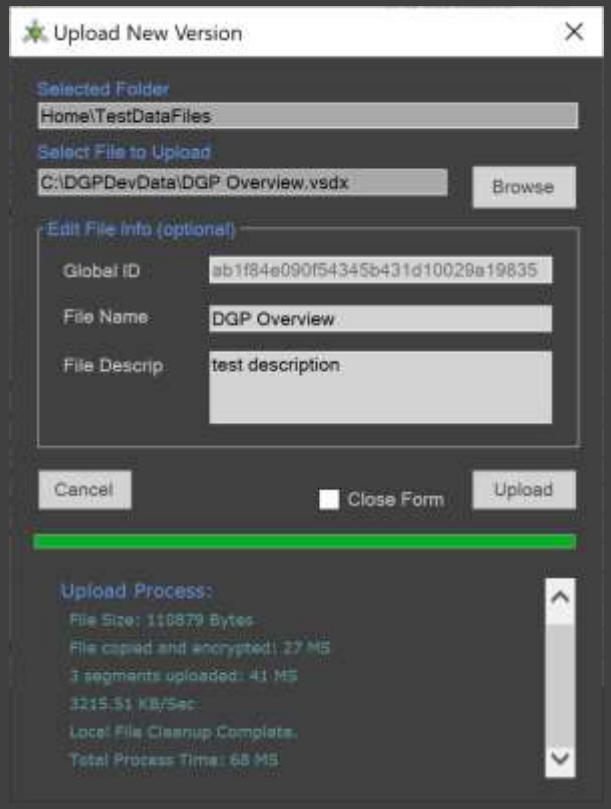
checked the Open checkbox, the downloaded file will be automatically opened using the application associated with the file extension by Windows (if no associated app is found, it will prompt the user to select an application).

If the file hash values do not match, the downloaded file is deleted and the error is displayed to the user.

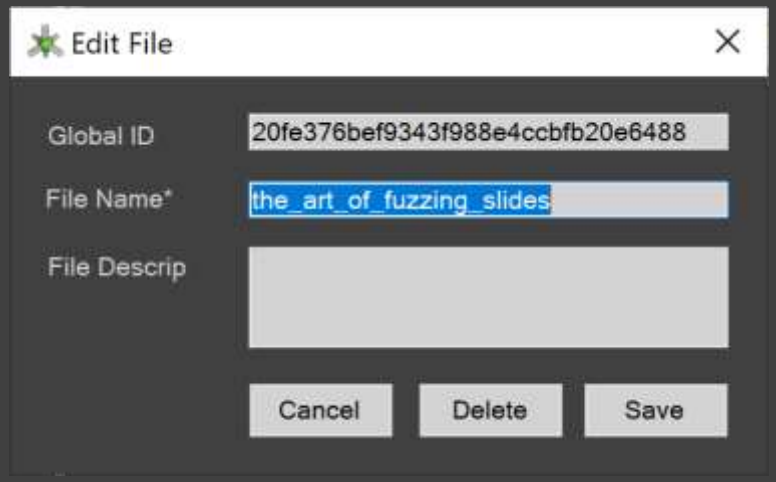
File Context Menu	Description
File Tags…	Displays the File Tags dialog, which enables the assignment of tags to a file. Tags are assigned to a file by double-clicking on the record in the Tag List grid. Tags are removed from files by double-clicking the record in the File Tags grid.



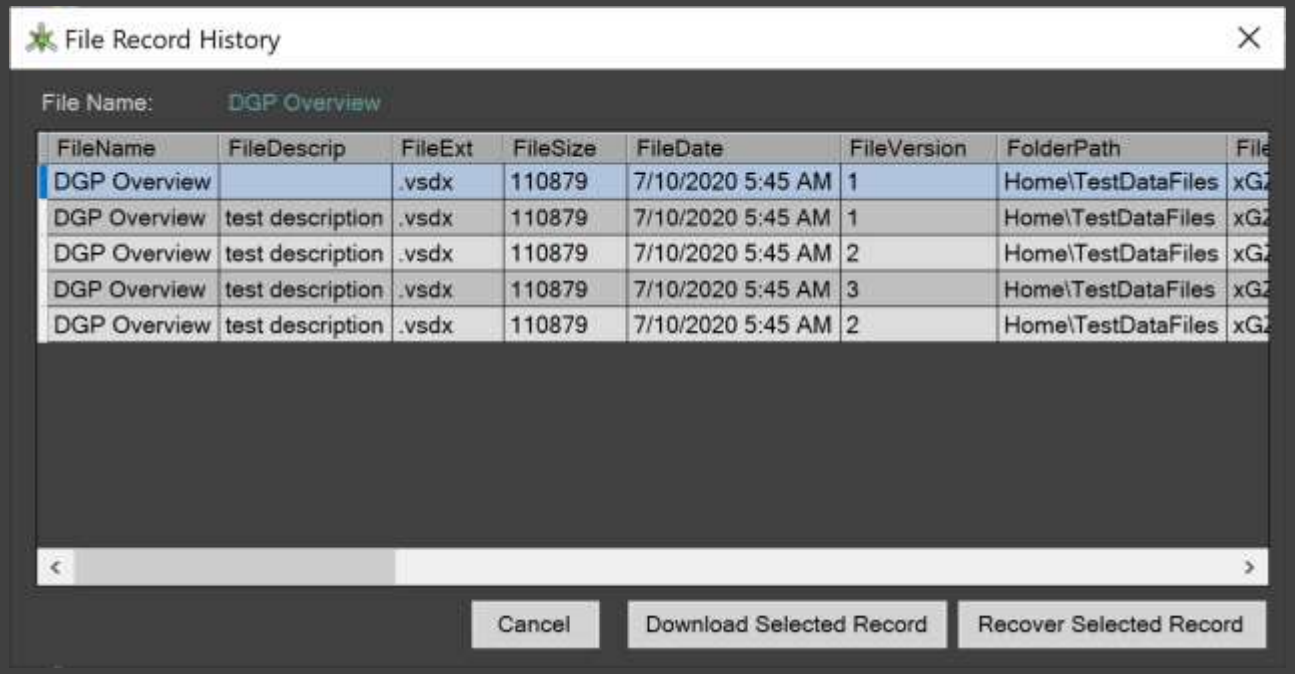
File Context Menu	Description
Upload New Version…	Displays the Upload New Version dialog which allows the user to upload a new version of an existing file (this process is slightly different from adding a new file to a system for the first time). The process collects the file data of the selected file, calculates the new hash value, increments the file version number, and uploads the file segments as FileShard records in one of the writable shards. When that process is completed, it updates the existing File record with the incremented version number and the new file info, hash value, etc.



File Context Menu	Description
Edit File Info…	Displays the Edit File dialog which allows the file name and file description values in the file info record to be edited. This creates a new version of the file info record only (no file upload), and so these types of edits are replicated as normal replica schema records.



File Context Menu	Description
File History…	Displays the File Record History dialog which shows all versions of the selected file. Archived versions of the file can be downloaded using the File Download dialog. If a user wishes to "rollback" changes made to a file, they can recover a selected archived version, which will create a new file info record to supersede the current active version. This type of change only modifies the file info record and does not require a new file to be uploaded.



File Context Menu	Description
Remove From Favorites	Only shown when the file grid was populated by clicking on the Favorites folder. Selecting the menu item will remove the file from the user's favorites list.
Add To Favorites	Displayed when the file grid is populated by any search other than clicking the Favorites folder. Selecting the menu item will add a link to the file to the user's favorites list.

Copy and Move Files

Due to the complexity of merge replication and the different access levels of folders, copying and moving files is currently a manual process.

To copy a file to a new folder, download the file and then upload it to the new folder. Currently, this is the only practical way to ensure that the new folder path is correct and that the account has read-write access to the new folder location in the folder tree.

Moving a file follows the same process as copying a file, with the added final step of deleting the file from its original folder.

To prevent file transfers from being used as a mechanism for Denial of Service attacks, users are currently restricted to uploading and downloading a single file at a time. While it is not technically difficult to batch uploads and downloads, finding ways to limit the size of the batch operations (in order to prevent DDOS attacks) when copying potentially large numbers of files is somewhat difficult.

Drag-and-drop along with batch operations of file copies, moves and deletes are planned for a future release, but will require a queueing mechanism to be added to the Lattice UI.

Tags Functionality

Tags are maintained using the Tags form. The combination of TagName and TagDescrip define each unique tag, so it is possible to have multiple tags with the same TagName, as long as the descriptions are different.