

Junk Finder v5

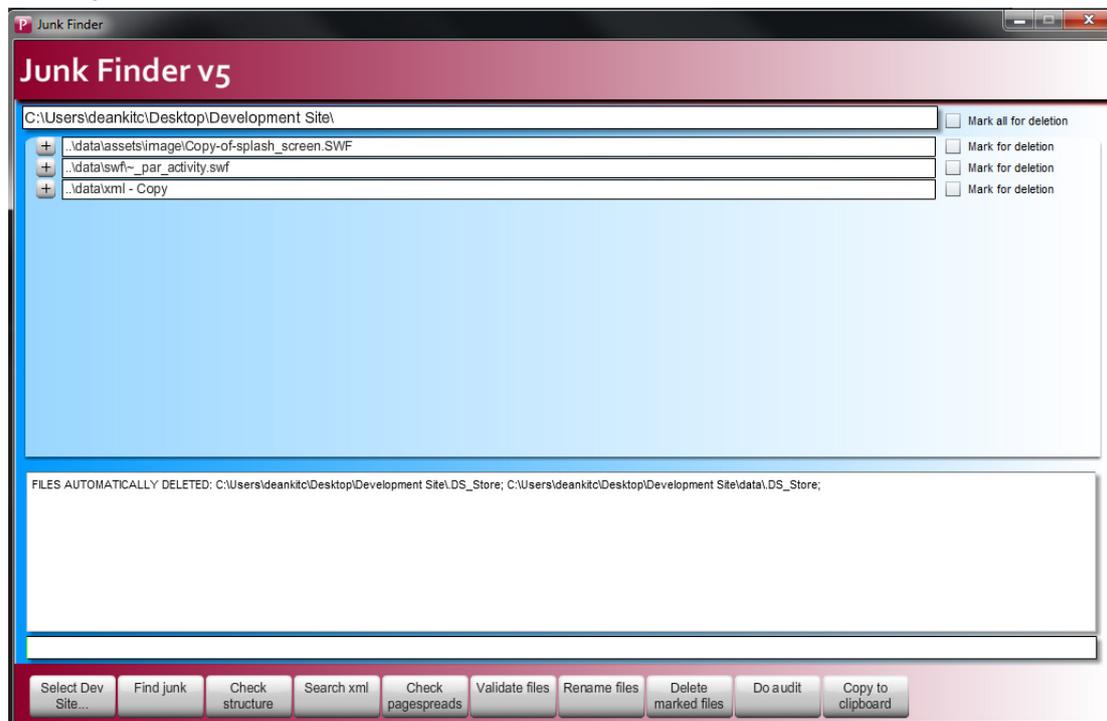
BACKUP THE DEVELOPMENT SITE BEFORE RUNNING THE JUNK FINDER TOOL.

SELECT DEV SITE.....	1
FIND JUNK	1
CHECK STRUCTURE.....	2
SEARCH XML	3
CHECK PAGESPREADS	4
VALIDATE FILES	4
RENAME FILES.....	5
DELETE MARKED FILES	6
DO AUDIT	6
COPY TO CLIPBOARD	6

Select Dev Site...

Use this button to select the root of a development site. The Junk Finder checks for the existence of a main.swf file and uses it's location to determine if the development site is AT Platform 2.0, 2.1, Basic, Hybrid or Legacy AS2. A different development site can be selected without closing / reopening the tool.

Find junk

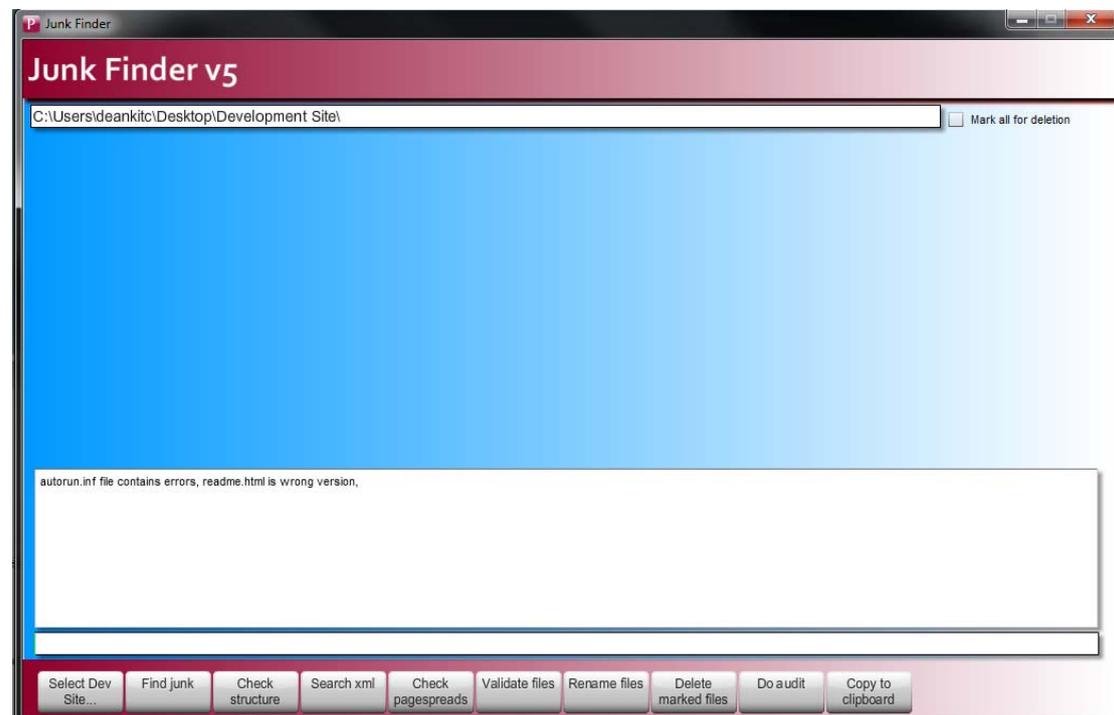


This lists any files and folders across the entire development site which are considered 'junk'. The _settings.xml file contains a pre-defined list of what file /

folder strings are flagged as junk. It also contains a list of files which are excluded from the list, which may otherwise have been picked up by the search criteria. The '+' button next to the file will open the directory the file is contained in. A file can be marked for deletion using the 'Mark for deletion' checkbox. The 'Mark all for deletion' checkbox will toggle selection of all files on / off. Once selected, files marked for deletion are highlighted in red.

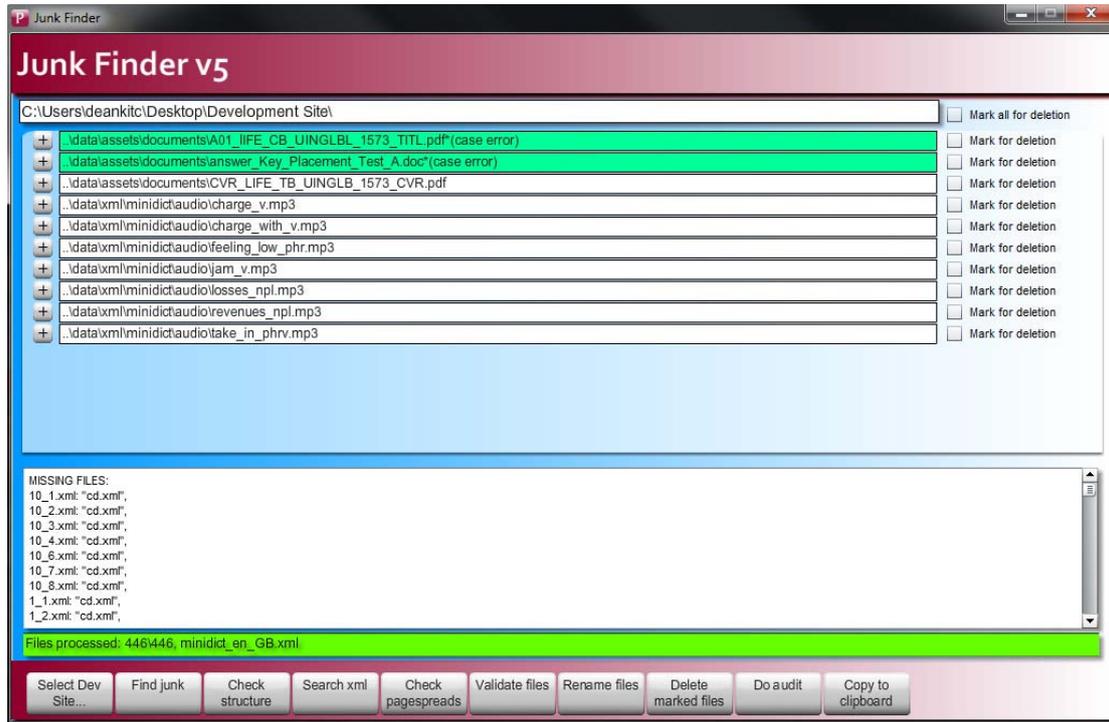
The bottom text box contains a list of files which have been automatically deleted by the Junk Finder. These will consist of MAC temporary files such as .DS_Store. If these files are left in a development site, they may get picked up when creating an installer manifest and then subsequently will cause the installer to fail.

Check Structure



Since the Junk Finder knows the type of development site based on the main.swf location, the Check Structure button will get it to check the contents of the development site to make sure key files are present, and their contents is correct. The _settings.xml file contains lists of what are considered to be key files and/or folders and where they should appear in the development site structure. These are case-sensitive. If the development site doesn't contain these files in the location specified, it will flag as an error in the output text box. In addition, the contents of 'autorun.bat' and 'autorun.inf' are verified. This is done against template files found in the 'asset' folder of the Junk Finder. Each file needs to be an exact match for the template file otherwise it will get flagged as incorrect. The version number in the readme.html and _installer.xml file are also checked against the version number in the _settings.xml to ensure the development site has the correct versions of both. The vendor pack will contain the latest versions of all these files.

Search XML

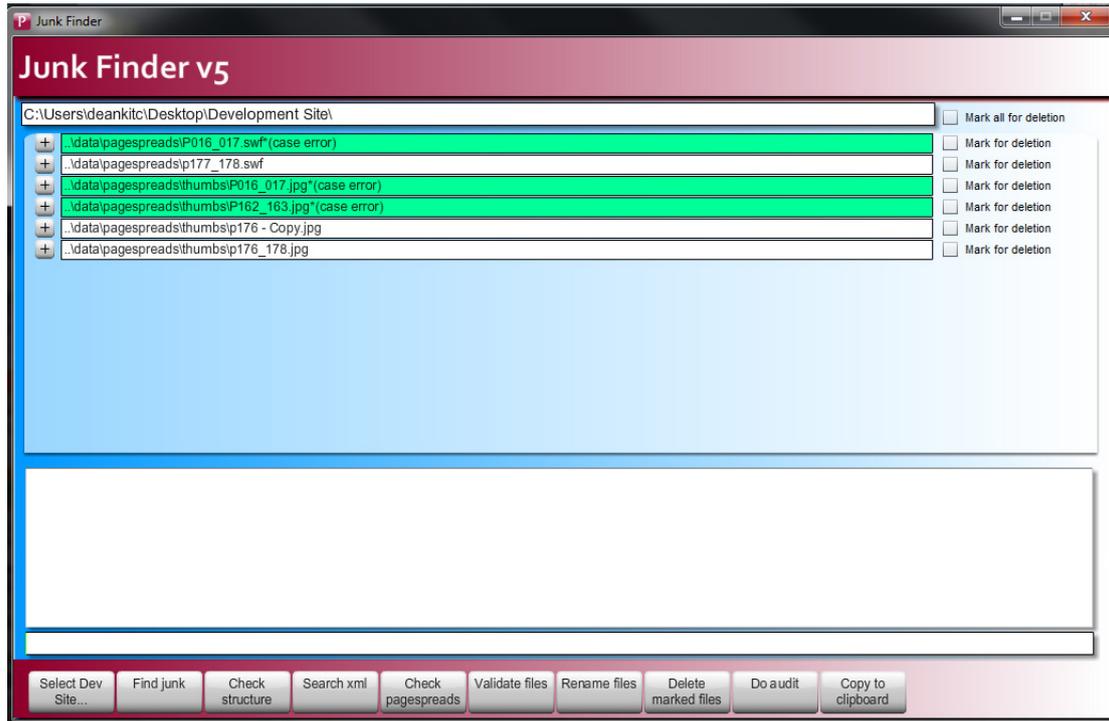


The top list contains all the physical asset files which appear in the asset folder / subfolders of the development site but are not referenced in an xml file contained in the xml folder (or its subfolders). The `_settings.xml` file contains a list of files which are excluded from the search results. Any files which are found to be referenced in xml but have a case error are flagged in green. Such files will cause problems when running a product on a case-sensitive filesystem such as that found on Linux.

The text box at the bottom contains a list of errors, for example any xml files which have parsing errors. Any such files are skipped during the search, so parsing errors should be corrected manually and the search xml re-run so accurate results can be returned. The text box also lists missing files. These are file references which exist in an xml file but for which a physical file cannot be found. This check is done across the entire development site. The list is organised by xml file, and a list of each file reference contained within that xml file which a matching file cannot be found. One limitation of this is that file-paths within xml will not be absolute so if a matching file exists but is in the wrong folder it will not be flagged by the search. This search will also display case errors and suffix the filename with `'*(case error)'`. This search works across mini-dictionary files too. The `_settings.xml` contains a tag `<filenameExtractionExtensions>` which defines which file types references are searched for within an xml file. Any file with an extension not defined in the `_settings.xml` will be excluded from the search results.

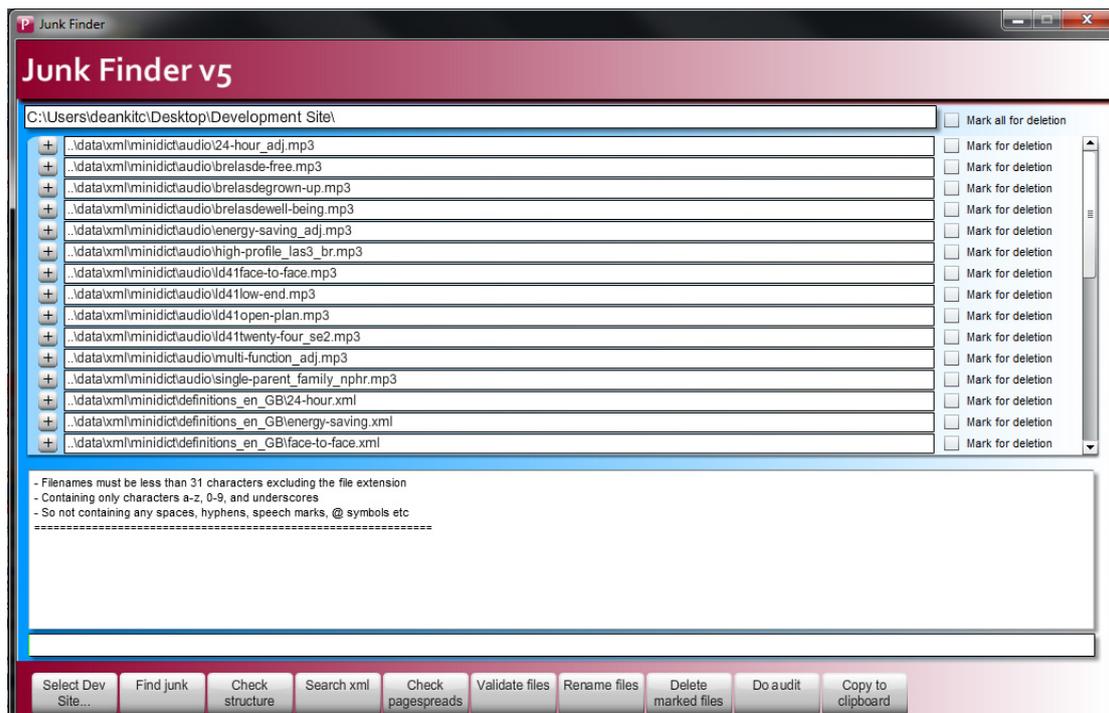
During the search, the green progress bar tracks which xml file is currently being processed. If for whatever reason the tool gets stuck processing a file, the filename will be visible in the progress bar.

Check Pagespreads



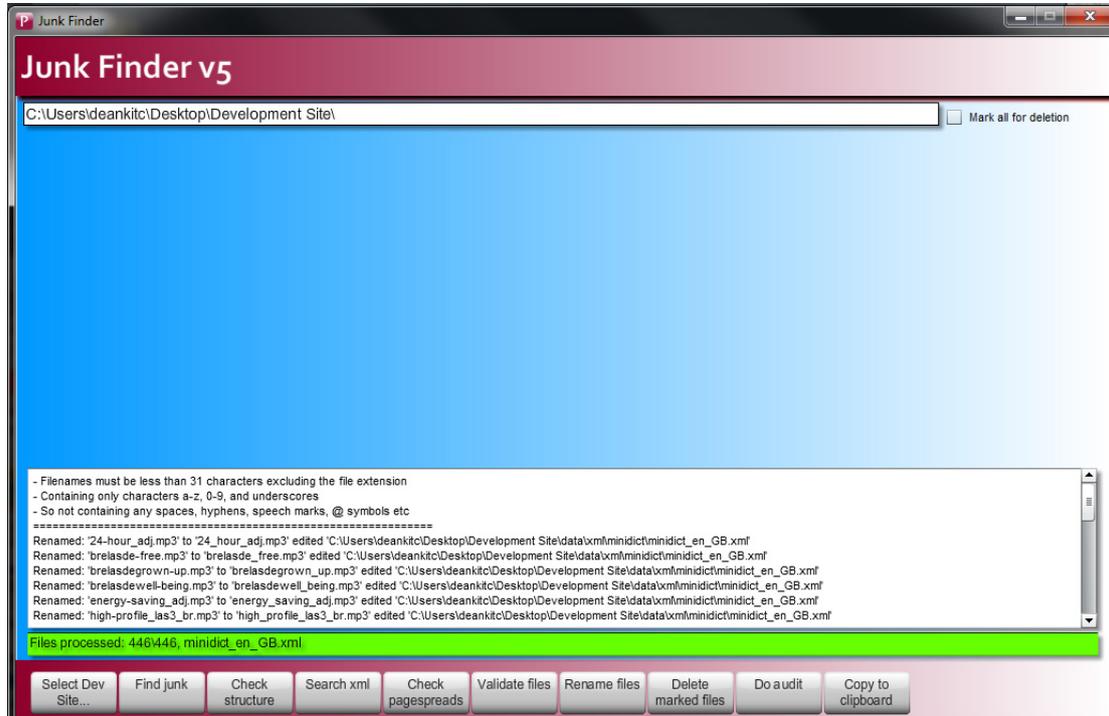
The Check Pagespreads search returns a list of pagespreads and thumbnails which physically exist in the development site but which are not referenced in the `_xml_navigation.xml` file. As with the 'Search xml' results, case errors are flagged in green. Note the 'Search XML' button will list any pagespread xml references for which a matching pagespread file cannot be found. But it won't list physical pagespread files which are not xml referenced as they are not within the asset folder / subfolders of the development site.

Validate files



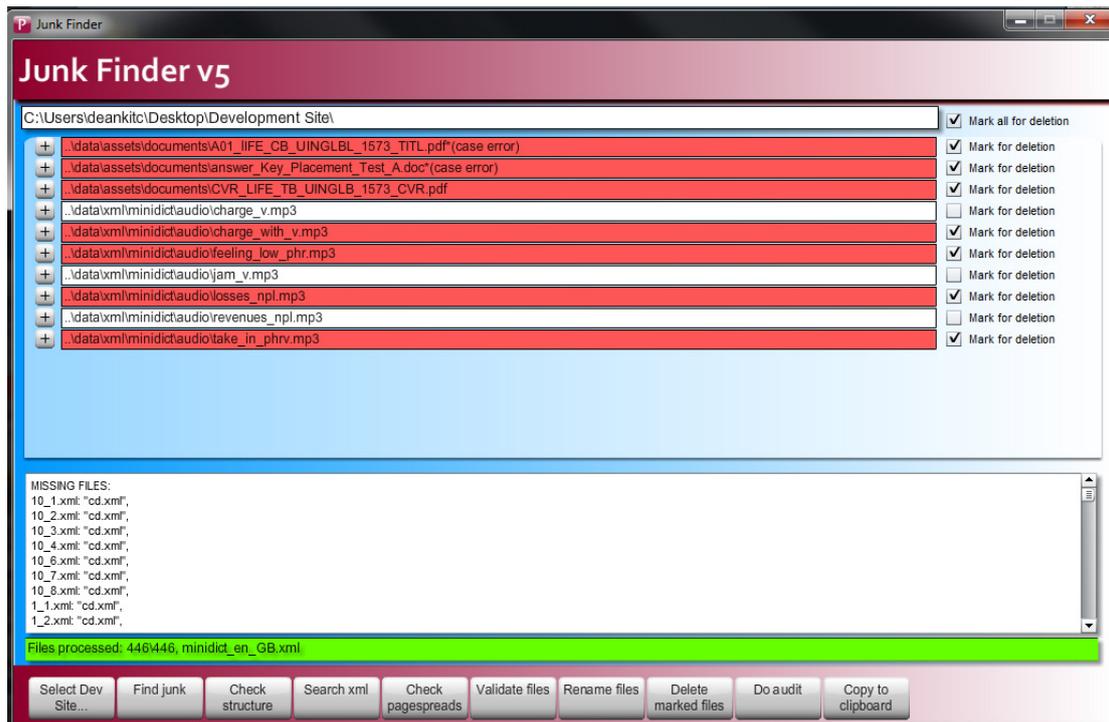
This lists files in the development site data folder whose filenames are not cross-platform friendly.

Rename Files



This automatically renames files in the development site data folder whose filenames are not cross-platform friendly, and updates the file's xml reference if it exists. The text box contains a report which lists the original filename, the new filename and the xml file it has edited. As xml files may need editing, sufficient administrator rights including write access to the xml files is necessary. After running the Rename Files process, the Search xml check should be run again to confirm if the automatic renaming has been successful.

Delete Marked files



Files which will be deleted when pressing the 'Delete marked files' button are highlighted in red. Please make sure the development is backed up before deleting anything as the files will be deleted and cannot be retrieved from the recycle bin etc.

Do Audit

This creates a hidden file called dir.log. A confirmation will appear with the save location of this file once it has been created. This log file can be opened in a text editor and contains details about all the files within the development site. It can be used for troubleshooting purposes by technical support after a product has been released to market. This log should only be created once the development site has been finalised and no more changes are to be made. For every file in the development site it will list the filename, full directory path, the file attributes (read only / hidden etc), the date it was created on and last time it was modified, and the file-size in bytes.

Copy to Clipboard

This copies the currently displayed results (both the filenames list and contents of the text box) to the system clipboard. They can be pasted into a text editor. The text will contain appropriate headers to make it clear what the results contain; file-paths are all truncated relative to the root of the development site.