**How AmazeFileManager works**

[LongClick 1](#_Toc528993383)

[com/amaze/filemanager/adapters/RecyclerAdapter.java 1](#_Toc528993384)

[folderSize 2](#_Toc528993385)

[com/amaze/filemanager/utils/files/FileUtils.java 2](#_Toc528993386)

[com/amaze/filemanager/asynchronous/asynctasks/CountItemsOrAndSizeTask.java 2](#_Toc528993387)

[AmazeFileManager\app\src\main\java\com\amaze\filemanager\asynchronous\asynctasks\LoadFolderSpaceDataTask.java 2](#_Toc528993388)

[com/amaze/filemanager/fragments/PropertiesSheet.java 3](#_Toc528993389)

[Show size 4](#_Toc528993390)

[com/amaze/filemanager/fragments/MainFragment.java 4](#_Toc528993391)

[com/amaze/filemanager/asynchronous/asynctasks/LoadFilesListTask.java 5](#_Toc528993392)

[com/amaze/filemanager/adapters/RecyclerAdapter.java 5](#_Toc528993393)

[PropertiesDialog 8](#_Toc528993394)

[com/amaze/filemanager/ui/ItemPopupMenu.java 8](#_Toc528993395)

[Sort 9](#_Toc528993396)

[com/amaze/filemanager/activities/MainActivity.java 9](#_Toc528993397)

[com/amaze/filemanager/ui/dialogs/GeneralDialogCreation.java 9](#_Toc528993398)

[com/amaze/filemanager/fragments/MainFragment.java 9](#_Toc528993399)

[com/amaze/filemanager/asynchronous/asynctasks/LoadFilesListTask.java 10](#_Toc528993400)

[com/amaze/filemanager/utils/files/FileListSorter.java 10](#_Toc528993401)

# LongClick

## com/amaze/filemanager/adapters/RecyclerAdapter.java

public void onBindViewHolder(final RecyclerView.ViewHolder vholder, int p) {

…

holder.rl.setOnLongClickListener(p1 -> {

// check if the item on which action is performed is not the first {goback} item

if (!isBackButton) {

toggleChecked(vholder.getAdapterPosition(), holder.checkImageView);

…

# folderSize

## com/amaze/filemanager/utils/files/FileUtils.java

public static long[] getSpaces(HybridFile hFile, Context context, final OnProgressUpdate<Long[]> updateState) {

…

if (hFile.isDirectory(context)) {  
 fileSize = hFile.folderSize(context);

## com/amaze/filemanager/asynchronous/asynctasks/CountItemsOrAndSizeTask.java

protected String doInBackground(Void[] params) {

…

if(isStorage) {  
 folderSize = file.getUsableSpace();  
} else {  
 folderSize = FileUtils.folderSize(file, data -> publishProgress(new Pair<>(folderLength, data)));  
}

## AmazeFileManager\app\src\main\java\com\amaze\filemanager\asynchronous\asynctasks\LoadFolderSpaceDataTask.java

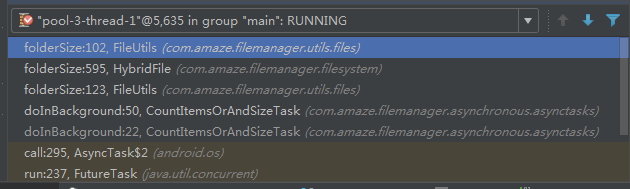
protected Pair<String, List<PieEntry>> doInBackground(Void... params) {  
 long[] dataArray = FileUtils.*getSpaces*(file, context, this::publishProgress);

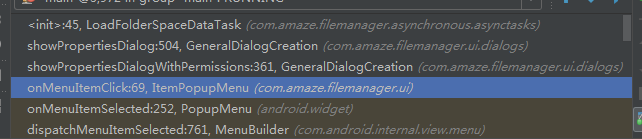
## com/amaze/filemanager/fragments/PropertiesSheet.java

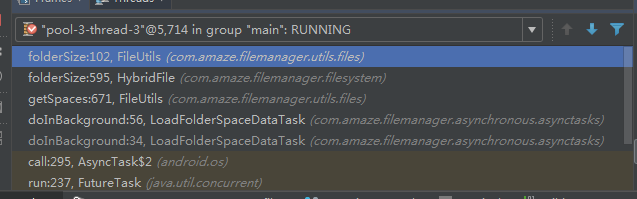
public void setupDialog(Dialog dialog, int style) {

…

mFileSizeTextView.setText(Formatter.*formatFileSize*(dialog.getContext(), mFile.isDirectory() ? FileUtils.*folderSize*(new File(mFile.getPath()), null) : mFile.getSize()));







# Show size

## com/amaze/filemanager/fragments/MainFragment.java

public void reloadListElements(boolean back, boolean results, boolean grid) {

…

if (adapter == null) {  
 adapter = new RecyclerAdapter(getMainActivity(), ma, utilsProvider, sharedPref,  
 listView, LIST\_ELEMENTS, ma.getActivity());  
} else {  
 adapter.setItems(listView, new ArrayList<>(LIST\_ELEMENTS));  
}

…

// method to add search result entry to the LIST\_ELEMENT arrayList  
private LayoutElementParcelable addTo(HybridFileParcelable mFile) {

…

if (mFile.isDirectory()) {  
 size = "-1B";

*/\*\*  
 \* This loads a path into the MainFragment.  
 \** ***@param*** *path the path to be loaded  
 \** ***@param*** *back if we're coming back from any directory and want the scroll to be restored  
 \** ***@param*** *openMode the mode in which the directory should be opened  
 \*/*public void loadlist(final String path, final boolean back, final OpenMode openMode) {

…

loadFilesListTask = new LoadFilesListTask(ma.getActivity(), path, ma, openMode,

…

## com/amaze/filemanager/asynchronous/asynctasks/LoadFilesListTask.java

protected Pair<OpenMode, ArrayList<LayoutElementParcelable>> doInBackground(Void... p) {

…

RootHelper.*getFiles*(path, ma.getMainActivity().isRootExplorer(), showHiddenFiles,  
 mode -> openmode = mode, file -> {  
 LayoutElementParcelable elem = createListParcelables(file);  
 if(elem != null) list.add(elem);  
 });

…

private LayoutElementParcelable createListParcelables(HybridFileParcelable baseFile) {

…

if (baseFile.isDirectory()) {  
 ma.folder\_count++;

## com/amaze/filemanager/adapters/RecyclerAdapter.java

public void onBindViewHolder(final RecyclerView.ViewHolder vholder, int p) { if (getBoolean(PREFERENCE\_SHOW\_LAST\_MODIFIED)) {

holder.date.setText(rowItem.date1);

} else {

holder.date.setVisibility(View.GONE);

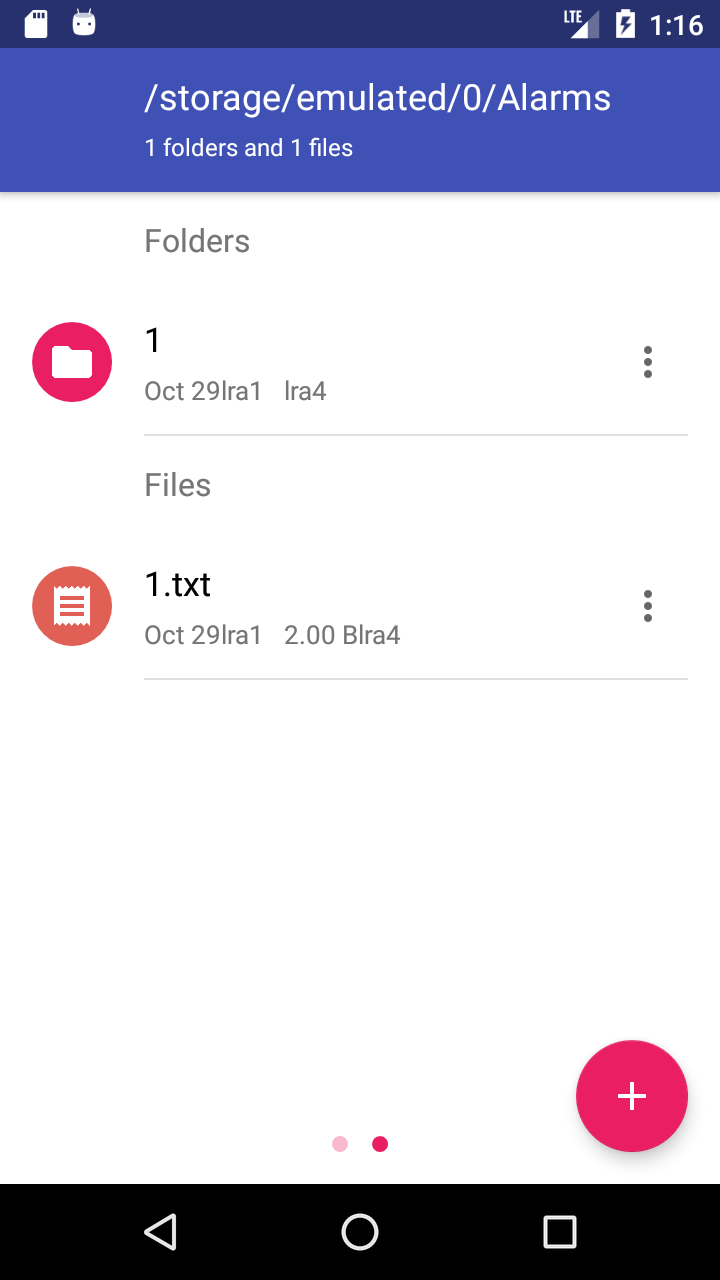
}

if (isBackButton) {  
 holder.date.setText(rowItem.size);  
 holder.txtDesc.setText("");  
} else if (getBoolean(*PREFERENCE\_SHOW\_FILE\_SIZE*)) {  
 holder.txtDesc.setText(rowItem.size);  
}

…

if (getBoolean(*PREFERENCE\_SHOW\_PERMISSIONS*))  
 holder.perm.setText(rowItem.permissions);  
 holder.date.setText(rowItem.date1+"lra1");  
} else {  
 holder.date.setVisibility(View.*GONE*);  
 if (getBoolean(*PREFERENCE\_SHOW\_LAST\_MODIFIED*)) {  
}  
  
if (isBackButton) {  
 holder.date.setText(rowItem.size+"lra2");  
 holder.txtDesc.setText("lra3");  
} else if (getBoolean(*PREFERENCE\_SHOW\_FILE\_SIZE*)) {  
 holder.txtDesc.setText(rowItem.size+"lra4");  
}

…



public void createHeaders(boolean invalidate, List<IconDataParcelable> uris) {

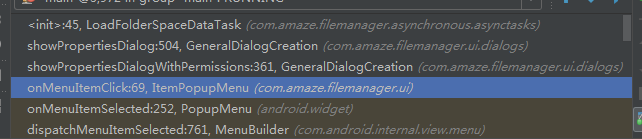
private void setItems(RecyclerView recyclerView, ArrayList<LayoutElementParcelable> arrayList, boolean invalidate) {

# PropertiesDialog

## com/amaze/filemanager/ui/ItemPopupMenu.java

public boolean onMenuItemClick(MenuItem item) {  
 switch (item.getItemId()) {  
 case R.id.*about*:  
 GeneralDialogCreation.*showPropertiesDialogWithPermissions*((rowItem).generateBaseFile(),  
 rowItem.permissions, (ThemedActivity) mainFragment.getActivity(),  
 mainActivity.isRootExplorer(), utilitiesProvider.getAppTheme());

…



# Sort

## com/amaze/filemanager/activities/MainActivity.java

case R.id.*sort*:  
 Fragment fragment = getFragmentAtFrame();  
 if (fragment instanceof AppsListFragment) {  
 GeneralDialogCreation.*showSortDialog*((AppsListFragment) fragment, getAppTheme());  
 }  
 break;  
case R.id.*sortby*:  
 if (ma != null)  
 GeneralDialogCreation.*showSortDialog*(ma, getAppTheme(), getPrefs());  
 break;  
case R.id.*dsort*:  
 if (ma == null) return super.onOptionsItemSelected(item);  
 String[] sort = getResources().getStringArray(R.array.*directorysortmode*);  
 MaterialDialog.Builder builder = new MaterialDialog.Builder(mainActivity);

## com/amaze/filemanager/ui/dialogs/GeneralDialogCreation.java

private static void onSortTypeSelected(MainFragment m, SharedPreferences sharedPref, Set<String> onlyThisFloders, MaterialDialog dialog, boolean desc) {

## com/amaze/filemanager/fragments/MainFragment.java

*/\*\*  
 \* This loads a path into the MainFragment.  
 \** ***@param*** *path the path to be loaded  
 \** ***@param*** *back if we're coming back from any directory and want the scroll to be restored  
 \** ***@param*** *openMode the mode in which the directory should be opened  
 \*/*public void loadlist(final String path, final boolean back, final OpenMode openMode) {

…

loadFilesListTask.executeOnExecutor(AsyncTask.*THREAD\_POOL\_EXECUTOR*);

…

public void onSearchCompleted(final String query) {

…

Collections.sort(LIST\_ELEMENTS, new FileListSorter(dsort, sortby, asc));

## com/amaze/filemanager/asynchronous/asynctasks/LoadFilesListTask.java

protected Pair<OpenMode, ArrayList<LayoutElementParcelable>> doInBackground(Void... p) {

…

Collections.*sort*(list, new FileListSorter(ma.dsort, sortby, asc));

## com/amaze/filemanager/utils/files/FileListSorter.java

public int compare(LayoutElementParcelable file1, LayoutElementParcelable file2) {