LinkBackADP

Linked Versioned Backups

Last Modified

Tuesday, June 19, 2012 at 05:29:58 PM

Document Revision: 18

Information Page

TODO: License Information

TODO: Disclaimer

Table of Contents

1. What is LinkBackADP

LinkBackADP creates a snapshot of a directory based on a source and destination folder. As an introduction, assume that I desire to backup /a/home/ to /b/home/. After configuring LinkBackADP to know that it should backup from /a/home/ to /b/home/, the following steps are performed:

1. Find the most recent snapshot in /b/home/.
2. Create a new snapshot directory named yyyyMMdd-hhmmss.
3. Check each file in /a/home/ against a previous backup to determine if the file should be copied or linked to the previous snapshot.
4. Write the snapshot result.

If two files are different between snapshots, a copy is made of the file. Files that are the same are hard linked so that they do not use more space than is needed. To understand links, think of a file as having a directory entry which references where the data is really stored. There are two types of links, hard links and soft links. Soft links are also referred to as symbolic links.

A soft link uses a path to reference the location of another file or directory. A soft link, therefore, can easily reference a file on another file system. If the linked file is moved, however, the link is not moved to follow it. In the example shown below, file1 is assumed to exist and file2 will symbolically link to file1. Use “`ls -ali`” to see the link.

```
ln -s /path/to/file1 /path/to/file2
```

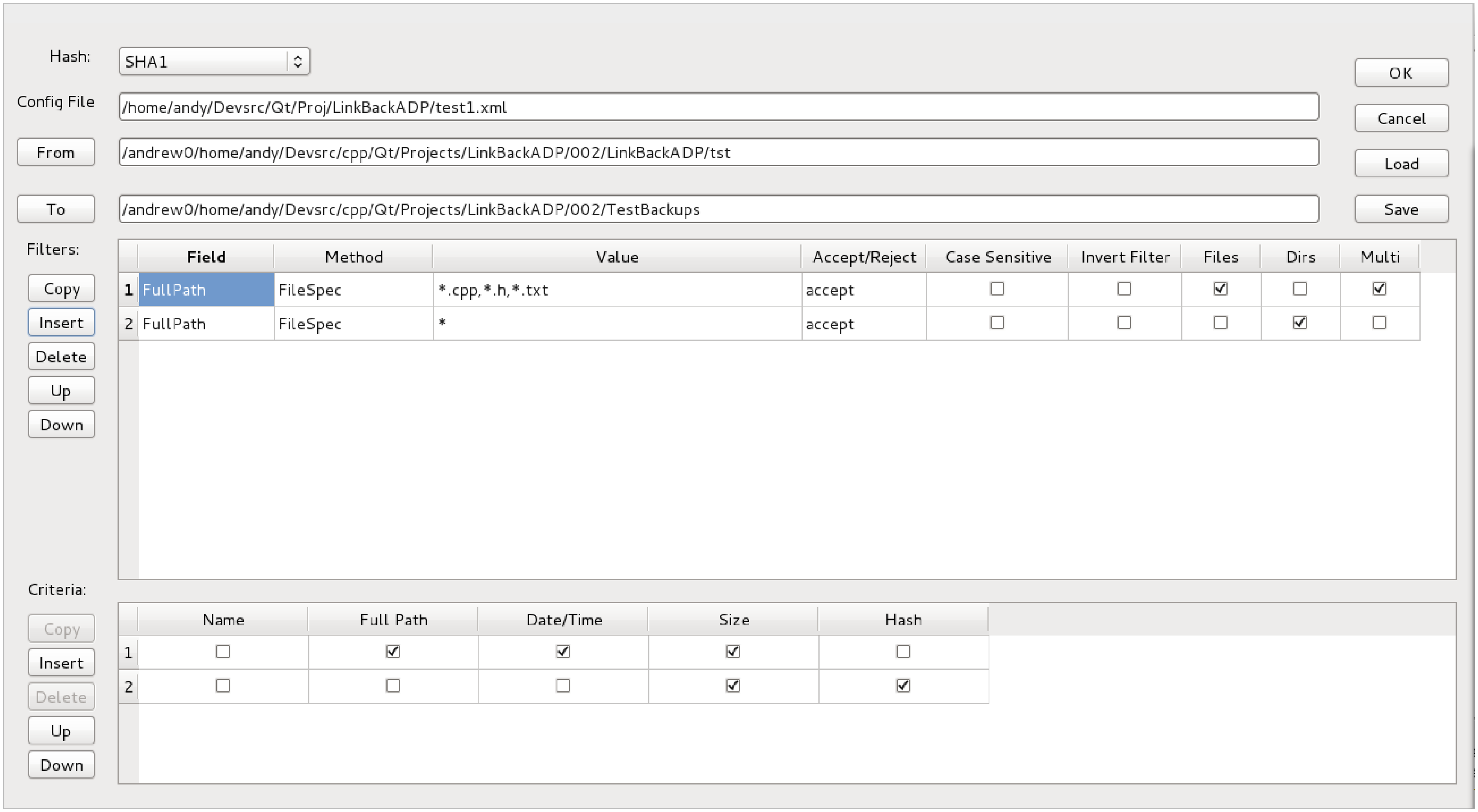
A hard link refers to the location of the physical data, so it cannot cross file system boundaries or link directories. Moving the file, however, will not affect a hard link because moving a file does not move the data itself. The example below creates a hard link as file2 to the existing file file1. If file1 is deleted, file2 is still valid. The data is not freed until all references to the data are removed.

```
ln /path/to/file1 /path/to/file2
```

This backup software creates hard links to files that are considered the same.

2. Configuration a backup

Use **Tools** > **Edit** to load the backup configuration dialog.



Hash: SHA1

Config File /home/andy/Devsrc/Qt/Proj/LinkBackADP/test1.xml

From /andrew0/home/andy/Devsrc/cpp/Qt/Projects/LinkBackADP/002/LinkBackADP/tst

To /andrew0/home/andy/Devsrc/cpp/Qt/Projects/LinkBackADP/002/TestBackups

Filters:

	Field	Method	Value	Accept/Reject	Case Sensitive	Invert Filter	Files	Dirs	Multi
1	FullPath	FileSpec	*.cpp,*.h,*.txt	accept	□	□	☑	□	☑
2	FullPath	FileSpec	*	accept	□	□	□	☑	□

Criteria:

	Name	Full Path	Date/Time	Size	Hash
1	□	☑	☑	☑	□
2	□	□	□	☑	☑

Figure 1. *Backup configuration dialog.*

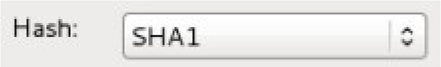
2.1. Hash

A hash function maps potentially large files to a smaller fixed length piece of data. A very simple hash for a string of characters might be to add the numerical (ASCII) value of each character modulo 256. This hash function can map all characters to a value between 0 and 256. The hash of “AUB” is 65 + 85 = 150, but so it the hash of “AAWWXZZZ” and “BT”.

The better the hash function, the less likely that two files will have the same hash value if the files are different. An MD5 hash uses 128 bits (16 bytes), so, if an MD5 hash is used, then every file is take down to 16 bytes and only those bytes are compared. The MD4 hash is the precursor to MD5, and it is included for Legacy reasons.

An SHA1 hash uses 160 bits (20 bytes), so it is marginally better than MD5, but collisions are still possible. There are also versions of SHA that return 256 and 512 bits, but they are not currently implemented in this software.

Is checking the file length and hash value good enough? Only you can decide. Forcing the file name to be the same as well, would allow you to catch files that have simply been moved.



Hash: SHA1

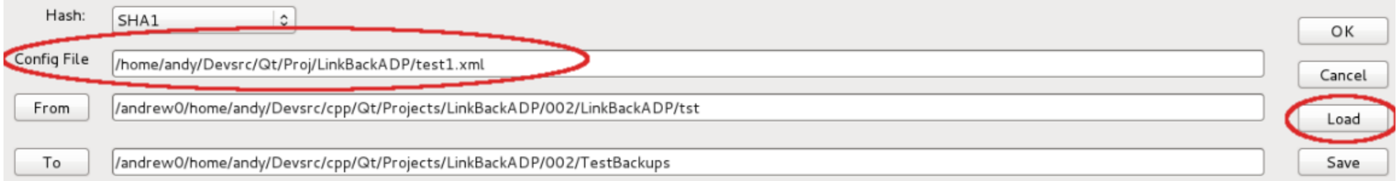
***Figure 2**. Select the hash used to determine if two files are the same.*

Caution	After a backup has been performed with a specific hash, all subsequent backups must use the same hash type because it is used to determine if two files are the same.

Select the desired Hash method from the drop list. The recommended value is populated when the software begins.

2.2. Specify a configuration file

Click the **Load** button to select an existing backup configuration file. The file name may be modified in the Config File text box. Use the **Save** button to save the configuration as an XML configuration file.



Hash: SHA1

Config File /home/andy/Devsrc/Qt/Proj/LinkBackADP/test1.xml

From /andrew0/home/andy/Devsrc/cpp/Qt/Projects/LinkBackADP/002/LinkBackADP/tst

To /andrew0/home/andy/Devsrc/cpp/Qt/Projects/LinkBackADP/002/TestBackups

***Figure 3**. Load a configuration file from disk.*

2.3. From and To locations

Click the **From** button to select the directory where backups will start; for example, if the From directory is /home, then the files an directories contained in /home will be backed up. This value may be edited directly in the text box.

Click the **To** button to select the directory where backups will be copied; for example, if the To directory is /archives, then the backup directories will begin in /archives. A new directory will be created in that specifies the date and time in the archives directory. For example, if the date and time is March 1, 2012 at 1:24:35 PM, then the new backup directory is /archives/20120301-132435. This new directory will contain a directory named “home” and a summary list of the files that are backed-up into that directory.

Caution	What directory is used if you try to backup from “/”?

Backups are created in a directory named from the directory name on which the backup is based. The code does not currently support an “empty” directory name.

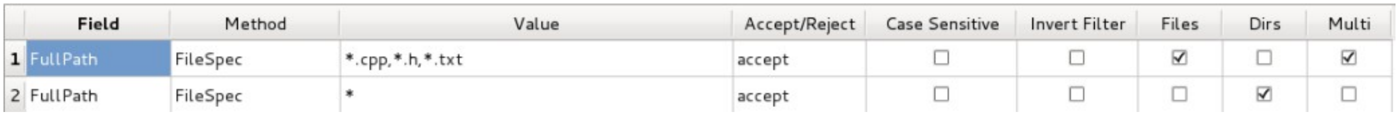
2.4. Filters: What files are ignored

The filters section determine which files are included in the backup and which files are ignored. The following columns are used

Table 1. *Filter fields.*

Column	Description
Field	Double click on the value in the field to change the text to a drop down list of valid file properties to compare against the value field. Valid values include the file date, date / time, time, full path including file name, file name only, path only, and file size.
Method	Double click on the value in the field to change the text to a drop down list of methods to compare the selected file properties (Field) to the entered Value. The following comparisons are supported: <, <=, =, >, >=, !=, Regular Expression, File Spec, and Contains.
Value	Enter the value used in the comparison. Check the Multi column if the value to compare is a comma separated list of values.
Accept / Reject	Each row can cause a file or directory to be accepted or rejected. Double click on the value to set this value to either accept or reject. If set to reject, then if a match is found, then that file is not considered for backup.
Case Sensitive	If checked, then comparisons are done in a case sensitive way, otherwise, comparisons are case insensitive.
Invert Filter	If checked, then the comparison results are inverted. As an example, to reject any file that does not contain the letters “abc” in the file name, set field to name, method to contains, value to “abc”, Accept/Reject to reject and then check invert filter. Any file name containing “abc” will match, but then that match is inverted. Note that this can also be done using a regular expression.
Files	Check if this filter is used to match files.
Dirs	Check if this filter is used to match directories.
Multi	Check if the value contains a comma delimited list of values; allowing multiple values to be specified in a single row.

Filters are processed one line at a time starting with the first filter (line 1).



	Field	Method	Value	Accept/Reject	Case Sensitive	Invert Filter	Files	Dirs	Multi
1	FullPath	FileSpec	*.cpp,*.h,*.txt	accept	□	□	☑	□	☑
2	FullPath	FileSpec	*	accept	□	□	□	☑	□

Figure 4. *Sample filters.*

The example shown in Figure 4 compares the entire path to each file to the FileSpecs *.cpp, *.h, and *.txt. Any file that matches in a case-insensitive way is accepted. If the **Multi** column were not checked, then the value field would be taken as a single value (as opposed to splitting values based on commas). No directory will match this filter because the filter only applies to directories.

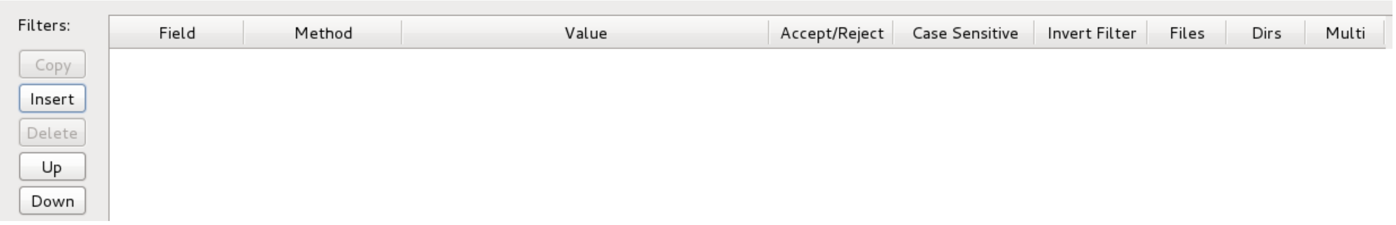
Any file or directory that does not match filter 1 is then compared against filter 2, which matches any directory (because it matches the file spec "*").

Any file or directory that does not match filter 2 is then rejected. The result is that all files with the extension cpp, h, or txt are included regardless of the directory.

In this example, using the full path for comparisons is probably wasteful because only the full name is of interest.

2.4.1. Create a filter

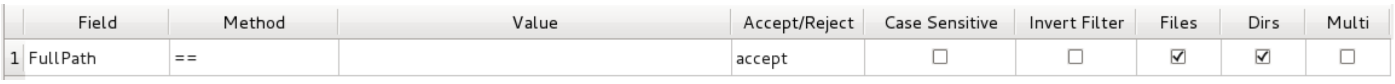
Use **Backup > Edit** to open the Backup configuration dialog. There are no filters present.



Field	Method	Value	Accept/Reject	Case Sensitive	Invert Filter	Files	Dirs	Multi

Figure 5. *There are no filters on initial start.*

Click **Insert** to add a new filter.

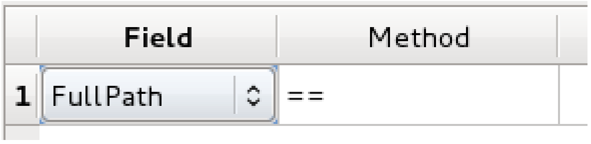


	Field	Method	Value	Accept/Reject	Case Sensitive	Invert Filter	Files	Dirs	Multi
1	FullPath	==		accept	☐	☐	☑	☑	☐

Figure 6. *Empty filter.*

Field

To change the field, double click in the Field column, and the column will change to a drop-down.



	Field	Method
1	FullPath	==

Figure 7. *Double click on the Field and it changes into a drop down.*

Click on the drop-down box to select the desired field on which to filter (see Figure 8).

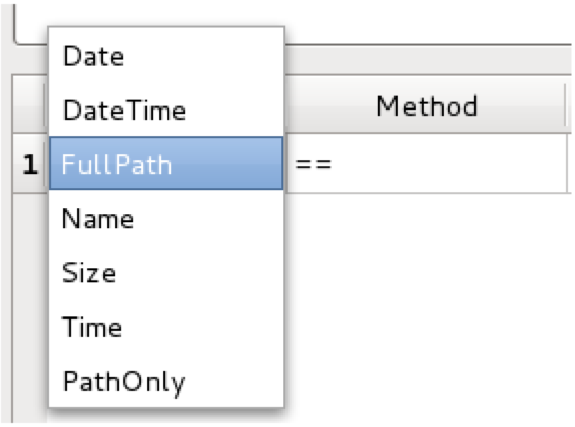


Figure 8*. Click on the drop-down to open the full list.*

Method

The Method column determines how the field specified in the Field column is compared against the Value column. The Method column acts similarly to the Field column; single click to create the drop-down and click again to open the drop-down and view the choices (see Figure 9).

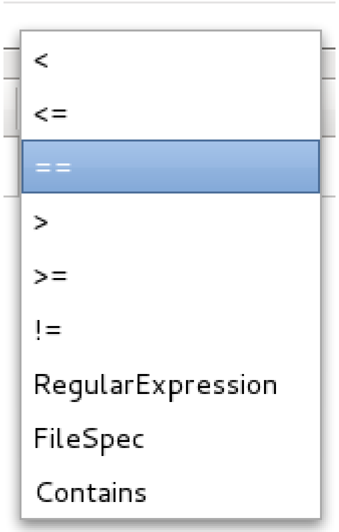


Figure 9*. Click on the drop-down to open the full list.*

As a reminder to those that do not regularly use regular expressions: A regular expression matches if a portion of the field matches the regular expression; if a full match is desired, be sure to place a “^” at the front and a “$” at the end of the regular expression.

Value

Enter the desired value to compare against the field column in the Value column. Multiple values can be entered in the Value column by separating the values with a comma and by checking the Multi column.

Accept / Reject

Set the value to accept or reject depending on whether the filter should cause a file or directory to be included in the backup (accept) or not included in the backup (reject). Double click on the value to change it to a drop-down box and then choose accept or reject.

Case Sensitive

Some operating systems use a case-sensitive comparison while searching for files and directories and others use a case-insensitive search. The default is for matches to be checked in a case-insensitive way. Check this column if you desire files to match using a case-sensitive match.

Invert Filter

Check Invert Filter to cause any file that is accepted / rejected by the filter to have the opposite action occur. As an example, to reject any file that does not contain the letter a, set the method to contains, the value to a, accept/reject to reject, and then set the invert filter to true. Note that this particular example can also be done using a regular expression.

Files

Check this column if the filter applies to files.

Dirs

Check this column if the filter applies to directories.

Multi

Check this column if the value field should be split on commas and it contains multiple values.

2.4.2. Organizing filters

Use the **Copy** button to copy an existing filter.

Use the **Delete** button to remove an existing filter.

Use the **Up** and **Down** buttons to move a filter up and down in the list.

Use the **Insert** button to insert a new filter into the list.

2.5. Criteria: When are files the same

If a file is the same as a file from a previous backup, I can create a link rather than making a new copy of the file. For my personal use, it is OK if I accidentally consider two files the same that are not a statistically small percentage of the time, that is acceptable for my home backups. I make my decisions based on my expectations of the data.

The Criteria section of the configuration dialog allows you to configure how the software decides if two files are equal.

***Table 2**. Criteria that can be used to determine if two files are the same.*

Criteria	Description
Name	The file name must match.
Full Path	The full path to the file, which includes the file name, must match.
Date / Time	File date and time must match.
Size	File size must match.
Hash	A hash of the file data must match.

You may enter multiple methods to determine if a file is the same as another. The first row is used to check for a match. If a match is not made, then the second row is used for a match, and so on. If all methods of determining if a file should match do not result in a match, then the file is copied rather than linked.

The first row as shown in the configuration dialog (see Figure 1) states that if the full path to the file, the file date and time, and the file size match that from a previous backup, then the files are considered the same. For my uses, this criteria matches most of the files in my backup, allowing me to quickly link over 100 GB of data files very fast. It is possible that a file was changed and then the date and time were purposely reset, then the files will erroneously match if the file size is the same.

The second line checks to see if there is a file from a previous backup that matches the file size and the file hash. If the answer is yes, the the files are considered the same and the files are linked rather than copied.

The intention is to use methods that will minimize the number of disk reads and writes.

3. Known issues

I initially sent a significant amount of output to the primary display, but the embedded QT control could not handle the load. Find another method to display this sort of thing to the user.

3.1. Todo

1. Handle a backup from the root directory “/” or “C:\”.
2. Support other fields such as “extension” or last changed date or time.
3. Better feedback while running.
4. Notice when the backup drive is running out of space and offer to remove a previous backup.