

Configure checksum checker

Files:

- [dspace/bin/checker](#)
- [dspace/config/dspace.cfg](#)

Instructions:

1. DSpace now comes with a Checksum Checker script ([dspace/bin/checker](#)) which can be scheduled to verify the checksum of every item within DSpace. Since DSpace calculates and records the checksum of every file submitted to it, this script is able to determine whether or not a file has been changed (either manually or by some sort of corruption or virus). The idea being that the earlier you can identify a file has changed, the more likely you'd be able to recover it (assuming it was not a wanted change).
2. There are several configuration options for the Checksum Checker which appear in the following section of dspace.cfg:

```
##### Checksum Checker Settings #####
The options you should most pay attention to are those regarding the checksum retention history (shown below). These two options specify how long a single checksum verification action is kept within your DSpace database. More information on each follows:
# check history retention
checker.retention.default=10y
checker.retention.CHECKSUM_MATCH=8w
```

3. The

```
checker.retention.CHECKSUM_MATCH
```

option specifies the timeframe after which a successful "match" will be removed from your DSpace database (defaults to 8 weeks). This means that after 8 weeks, all successful matches are automatically deleted from your database (in order to keep that database table from growing too large).

- The

```
checker.retention.default
```

option specifies the default timeframe after which all checksum checks are removed from the database (defaults to 10 years). This means that after 10 years, all successful or unsuccessful matches are removed from the database.

- You can specify any timeframe for either of these options. Valid timeframes include: seconds (s), minutes (m), hours (h), days (d), weeks (w), years 👍
 - Please note, these retention settings are only used if you specify the -p option for the checker (see below). Otherwise, they are ignored!
4. If you changed any option in the dspace.cfg, you will need to restart Tomcat (See Quick Restart in [Rebuild+DSpace](#)) for the changes to take affect.
 5. The Checksum Checker script ([dspace/bin/checker](#)) also has several command line options to be aware of:

- -c 10

= Limited Count Mode (

```
-c
```

). This limits the check to only the next 10 bitstreams (the checker will always start where it left off). (Recommended for larger repositories who may only want to check a portion of their repository each evening)

- -d 2h

= Duration Mode (

```
-d
```

). This tells the checker to run for 2 hours. The same timeframes are available as mentioned above for the dspace.cfg options. (Recommended for larger repositories who may only want to check a portion of their repository each evening)

- -b 111 112

= Specific Bitstreams Mode (

-b

). This tells the checker to only look at the bitstreams with internal IDs of 111 and 112.

- -a 1234/12

= Specific Handle Mode (

-a

). This tells the checker to only check bitstreams within the Item/Collection/Community specified by that handle.

- -l

= Looping Mode (

-l or -L

). A lowercase L (

-l

) specifies to check every bitstream in the repository once. (Recommended for smaller repositories who are able to loop through all their content in just a few hours maximum) An uppercase L (

-L

) specifies to continuously loop through the repository (not recommended for most repository systems)

- -p = Enable pruning (-p). Tells the checker to actually remove old results from the database based on the retention settings specified within dspace.cfg. Without this option, the retention settings are ignored and the database table may grow rather large!
6. You should schedule the Checksum Checker to run automatically, based on how frequently you backup your DSpace instance (and how long you keep those backups around for). The size of your repository is also a factor. For very large repositories, you may need to schedule it to run for an hour (e.g.

-d 1h

option) each evening to ensure it makes it through your entire repository within a week or so. Smaller repositories can likely get by with just running it weekly.

##*For Linux or Mac OSX, you can schedule it by adding a

cron

entry similar to the following to the crontab for the user who installed DSpace:0 4 * * 0 [dspace](#)/bin/checker -d2h -p

- (The above cron entry would schedule the checker to run scripts to run every Sunday at 4am for 2 hours. It also specifies to "prune" the database based on the retention settings in dspace.cfg. Note: You would need to change [dspace](#) to the full path of your DSpace installation directory.)
- For Windows, you will be unable to use the

checker

shell script. Instead, you should use Windows Scheduled Tasks to schedule the following command to run at the appropriate time(s):

' '[dspace]'/bin/dsrun.bat org.dspace.app.checker.ChecksumChecker -d2h -p

- (The above command should appear on a single line.)

7. Optionally, you may choose to receive automated emails listing the Checksum Checkers' results. There is no shell script for this functionality, but it's still a rather easy change. Just make sure to schedule it to run after the checker has completed its processing (otherwise the email may not contain all the results).
- There are a few options you can specify to this email script:

- -a

= send all results (everything specified below)

- -d

= only report on deleted bitstreams

- -m

= only report on missing bitstreams

- -c

= only report on bitstreams whose checksums changed

- -u

= only report on un-checked bitstreams

- You can also combine options (e.g.

-m -c

) for combined reports

- For Linux or Mac OSX, you can add another

cron

entry similar to the following to the crontab for the user who installed DSpace:

```
0 8 * * 0 '[dspace]'/bin/dsrun org.dspace.checker.DailyReportEmailer -a
```

- (The above cron entry would schedule the email to be sent at 8am each Sunday, reporting all the results. The above command should appear on a single line.)
- For Windows, you can use the same general command. However, you obviously should use Windows Scheduled Tasks to schedule it:

```
'[dspace]'/bin/dsrun.bat org.dspace.checker.DailyReportEmailer -a
```

- (The above command should appear on a single line.)