

Enable file cleaner

URL: <https://docs.grassfish.com/docs/enable-file-cleaner>

Archiviert am: 2025-07-17 18:30:52

The continuous operation of the server accumulates usage data. Grassfish provides a file cleaner to delete this data and avoid filling up the server file system.

To enable the file cleaner, perform the following steps:

1. Navigate to the master.config file.
2. Under **appSettings**, look for **DataGateway**.
3. Set the value of **FileCleaner:Enabled** to **true**.
4. Set the value of **FileCleaner:IntervalSeconds** to the desired number of seconds.
5. Set the value of **FileCleaner:ConfigFile** to the file path of the filecleaner.xml file.
6. Save your changes and close the file.

```
<appSettings type="DataGateway">
  <add key="FileCleaner:Enabled" value="true"/>
  <add key="FileCleaner:IntervalSeconds" value="120"/>
  <add key="FileCleaner:ConfigFile" value="\\grassfish.local\GFServers\configs\filecleaner.xml"/>
</appSettings>
```

Note

If you upgrade from IXM One version 11.14 or earlier, you must migrate the configuration settings from DataSpots to DataGateway and update the keys and values.

Clean up directories

Note

This feature requires IXM One version 12.0.0 or later.

In addition to cleaning files, the file cleaner can also clean empty directories. In this case, the files are removed first and then any remaining empty directories are deleted. This makes for a cleaner and more organized directory structure and reduces unnecessary clutter.

You can configure your custom cleanup settings in the filecleaner.xml file.

Every cleanup setting requires a `name` and the following config parameters:

- `extensions` *Required*

Default: `string.Empty`

Specify wildcards for file extensions to search for during cleanup.

Examples: `*.log`, `*.jpg`, `*`

- `directory` *Required*

Default: `string.Empty`

Specify the directory to search for files to clean. Use `id` as a placeholder for a numeric ID.

Examples: `\\path\to\main\folder\log`, `\\path\to\main\folder\data\id\screenShot`

- `olderThanDays` *Optional*

Default: `30`

Specify after how many days files are deleted.

- `recursive` *Optional*

Default: `false`

Set to true to scan sub-directories for files or empty directories to clean or delete them.

- `deleteEmptySubDirectories` *Optional*

Default: `false`

Set to true to delete empty sub-directories.

Example:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<itemList>
  <item name="Serverlogs">
    <extensions>*.log</extensions>
    <directory>\\path\to\main\folder\log</directory>
    <olderThanDays>10</olderThanDays>
    <recursive>true</recursive>
  </item>
  <item name="ClientLogs">
    <extensions>*.log</extensions>
    <directory>\\path\to\main\folder\data\logs</directory>
    <olderThanDays>30</olderThanDays>
    <recursive>true</recursive>
  </item>
  <item name="DataTemp">
    <extensions>*</extensions>
    <directory>\\path\to\main\folder\data\tmp</directory>
    <olderThanDays>5</olderThanDays>
    <recursive>true</recursive>
  </item>
  <item name="Export">
    <extensions>*.csv</extensions>
    <directory>\\path\to\main\folder\data\export</directory>
    <olderThanDays>60</olderThanDays>
    <recursive>true</recursive>
  </item>
  <item name="Screenshots">
    <extensions>*.jpg</extensions>
```

```
<directory>\\path\to\main\folder\data\${id$\screenShot</directory>
<olderThanDays>90</olderThanDays>
<recursive>true</recursive>
</item>
<item name="TempData">
  <extensions>*</extensions>
  <directory>C:\Users\ServiceUserDev\AppData\Local\Temp</directory>
  <olderThanDays>4</olderThanDays>
  <recursive>true</recursive>
  <deleteEmptySubDirectories>true</deleteEmptySubDirectories>
</item>
</itemList>
```