

Restricting File Upload Extensions in Lucee

Restricting the types of files that Lucee applications can upload is an important security measure that every server administrator should consider implementing. If an upload form is not properly secured, or an attacker exploits a vulnerability in Lucee or an application, they may attempt to upload an executable file such as `bad.cfm` or `bad.php`. If the uploaded file is stored in a web-accessible directory, the attacker could request it through the website and potentially execute commands, steal data, install malware, or use the server to attack other systems.

Lucee supports blocking upload extensions at several levels:

- Server-wide, using an environment variable upon startup
- Application-wide, using `Application.cfm/Application.cfc`

The server-wide blocklist provides a default blanket rule for all Lucee applications on the server so that everything is protected by default for the defined blocked extensions. If some applications need to override the server-wide extension block list, they can do so using their application-wide setting, when necessary.

Server-Wide File Extension Blocking

The `LUCEE_UPLOAD_BLOCKLIST` environment variable defines the default list of extensions blocked by Lucee file-upload operations, including `cfile` action="upload". Lucee does not have an option to only allow specific extensions, so be sure to block every type of extension you can think of that could be used inappropriately if uploaded.

You can also set a JVM flag instead of an environment variable; the correct flag to use is `-Dlucee.upload.blocklist=""`.

Platform Independent

Whether you are running Lucee on Windows or on Linux, you can implement your blocklist using the JVM flag approach, rather than the environment variable we'll cover momentarily. On your server, open the file responsible for setting the JVM flags. On Windows, this is usually `tomcat/bin/setenv.bat`, and on Linux this is usually `tomcat/bin/setenv.sh`. On Windows, you can also just launch the Lucee-Tomcat Service Control app from the Start Menu.

Locate the line responsible for the Java options. If editing the file directly, this would typically start with `CATALINA_OPTS` or `export CATALINA_OPTS`. In the list of Java flags, you would add the below (make sure the flag before and after the one you add has proper space so it does not break the config, and as always, be sure to back up the file before even making changes).

```
-Dlucee.upload.blocklist=""
```

Be sure to add the full list of blocked extensions, not just an empty list; otherwise nothing will be blocked. Be sure to restart the Lucee service after saving your desired change for this to take effect.

Lucee on Linux

For standard Lucee installations, you can edit the `setenv.sh` file in Tomcat's bin directory. For example, that might be: `/opt/lucee/tomcat/bin/setenv.sh`

Be sure to make a backup of the file before modifying, which you can do with the `cp` command: `cp /opt/lucee/tomcat/bin/setenv.sh /opt/lucee/tomcat/bin/setenv.sh.bak`

On a new line within the `setenv.sh` add the following (shown below) and make sure it is all on a single line (replace the extensions with the desired extensions you want to block):

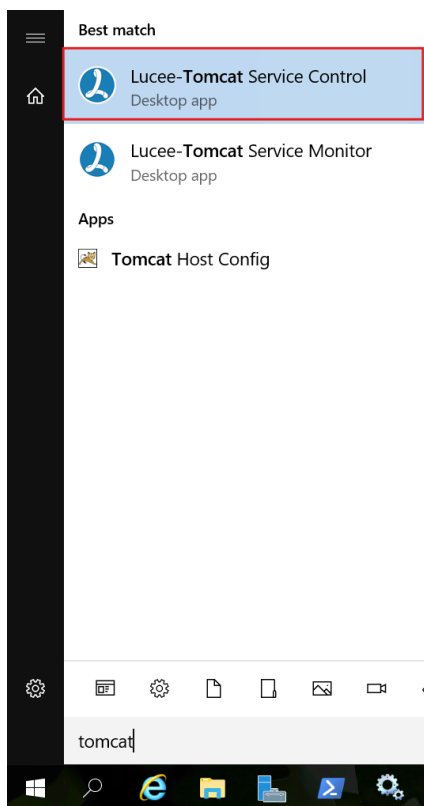
```
export LUCEE_UPLOAD_BLOCKLIST="asp,aspx,asa,asax,ascx,ashx,asmx,axd,bat,cfc,cfm,cfml,cgi,cmd,com,cshtml,dll,do,exe,htaccess,htpasswd,hta,inc,jar,json,jsp,jsp,lasso,mjs,msi,msp,php,php3,php4,php5,php7,php8,pht,phtml,phps,pl,ps1,ps1xml,ps2,ps2xml,psc1,psc2,py,pyc,pyw,rb,reg,scr,sh,shtml,vb,vbe,vbs,war,wsh,xap,xsl,xslt"
```

Again, make sure this is all on a single line otherwise it may cause issues when starting the Lucee application back up. Be sure to restart Lucee after saving the change for this to go into effect.

Lucee on Windows

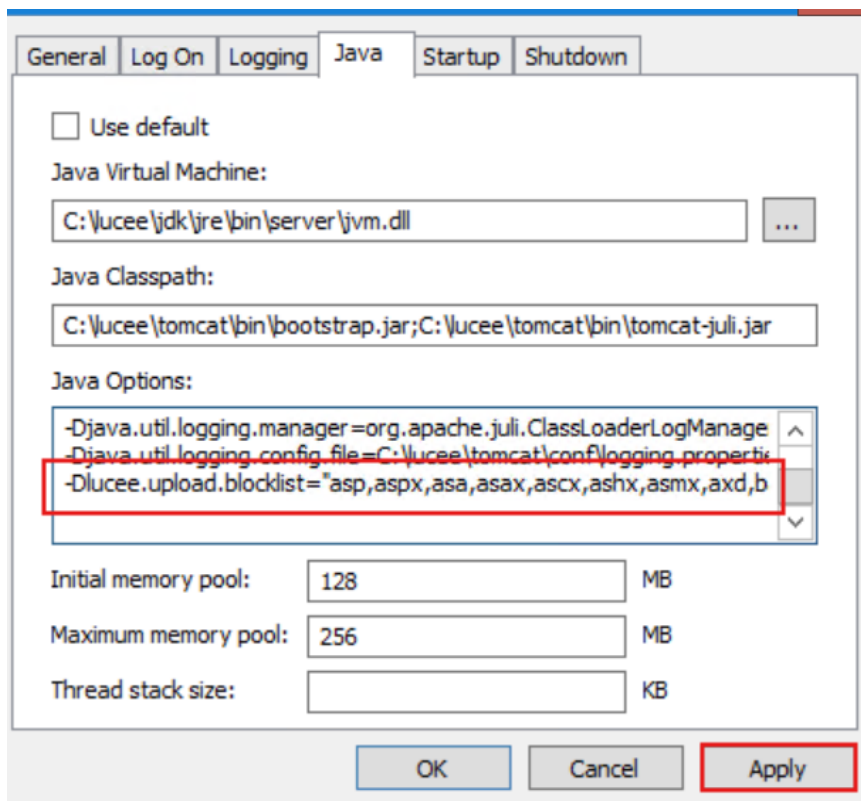
For standard Lucee installations, you can either edit the `setenv.bat` within the `tomcat/bin/` directory, or use the Lucee-Tomcat Service Control app. I always recommend using the service control app, so since we already showed the example above (for Lucee on Linux, which is the same when editing the file method), we will show an example of using the app method.

You can launch the app via the Start Menu after searching for "tomcat". The name should show something similar to "Lucee-Tomcat Service Control" unless changed, as shown in screenshot below:



You can also launch the Lucee-Tomcat Service Control app directly from the bin tomcat/bin/ directory: `C:\lucee\tomcat\bin\Luceew.exe`

Once the service control app is open, click on the "Java" tab, then on a new line by itself, add the Java flag mentioned earlier with your desired extension blocklist, as shown in the screenshot below:



Be sure to click the "Apply" button to save your changes. Then restart the Lucee service for this new change to take effect.

Application-Level Configuration

Regardless of the server-wide blocklist, you can define a custom extension blocklist on the application level. We recommend implementing a server-wide blocklist to protect the entire server and then when needed, to override on the application-level to allow something that is blocked on the server-wide level. This can be done using the websites `application.cfm` or `application.cfc` and defining the `blockedExtForFileUpload` for the application. If not set, then it will inherit the server-wide `blockedExtForFileUpload` setting.

Below is an example of implementing this in your `application.cfc`:

```
component {  
    this.name = "ExampleApplication";  
    this.blockedExtForFileUpload = "asp,aspx,cfc,cfm,cfml,exe,jsp,php,sh";  
}
```

Be sure that the `blockedExtForFileUpload` is all on a single line (excluding word-wrap in your editor) and your entire desired block list, as the extensions above are just examples.

Below is an example of implementing this in your `application.cfm`:

Override `blockedExtForFileUpload` on `application.cfm`

```
<cfapplication name="ExampleApplication" blockedExtForFileUpload="asp,aspx,cfc,cfm,cfml,exe,jsp,php,sh">
```

Be sure that the `blockedExtForFileUpload` is all on a single line (excluding word-wrap in your editor) and your entire desired block list, as the extensions above are just examples.

Note: When changing on the application level, you may need to either restart Lucee or restart your application (such as implementing a quick [ApplicationStop\(\)](#) in your code).

If you need any help with file upload extensions in Lucee, please [get in touch](#) with our Support team.

Related articles

- [Restricting File Upload Extensions in Lucee](#)
- [How to Reduce Your Backup Usage in R1Soft](#)
- [SmarterMail Lockdown Guide](#)
- [How to Grant Vivio Access to Your Cloudflare Account](#)
- [How to Restart Lucee on Windows and Linux](#)