

# How to check the content type of uploaded files

Since Blueriq 12.10, the runtime uses Apache Tika to determine whether the file extension matches the actual content of a file. This way, if a user would rename `virus.exe` to `document.pdf`, the runtime will detect that the uploaded file is not a PDF, and it will not continue with the process.

This feature is enabled by default. If needed, it can be disabled by setting [properties](#):

```
blueriq.fileupload.detect-content-type=false
```

```
blueriq.fileupload.validate-content-type=false
```

This will speed up the upload process, but introduces a security risk, so only use upon careful consideration.

## Office Documents

By default, Apache Tika will use some basic file scanning to detect the content type of a file. It does so by reading a few kilobytes from the start of the file, and search for some evidence, a so called [magic number](#), that defines what kind of file was uploaded. This mechanism is relatively fast, but is not fool proof. Attackers could be able to craft some files that contain a valid magic number, but are actually another file type.

As a result, this magic number scanning detects the new Office documents (docx, xlsx, pptx, etc.) as "office document". This means that you can upload a Word document with an xlsx extension, and the Runtime would accept that.

It is possible to add an additional Tika module to your custom Runtime, so that the contents of the Office documents and other documents are inspected more thoroughly. Note that the Runtime will need extra time to inspect the documents, but the inspection is more complete. This is a consideration that you should make for your solution.

To add Office document detection, you can add this dependency to your `dependencyManagement` section in your parent pom file:

### pom.xml

```
<dependencyManagement>
  <dependencies>
    ...
    <dependency>
      <groupId>org.apache.tika</groupId>
      <artifactId>tika-parsers-standard-package</artifactId>
      <version>${apache.tika.version}</version>
    </dependency>
    ...
  </dependencies>
</dependencyManagement>
```

And refer to the dependency in the dependencies section:

### pom.xml

```
<dependencies>
  ...
  <dependency>
    <groupId>org.apache.tika</groupId>
    <artifactId>tika-parsere-standard-package</artifactId>
    <scope>runtime</scope>
  </dependency>
  ...
</dependencies>
```

Note that the core module of Apache Tika is already included in the Runtime. Depending on the way you set the project up, you may have to redefine the `apache.tika.version` property to the same value as the version that is shipped with the runtime.

## Best effort

Apache Tika does content detection on an best effort basis, so there may be false positives or false negatives.