

# FileUploadWithAction

## File Upload with processing in an Action

The code below is tested with Cocoon 2.03. Refer to the end of the page if you want to use it with Cocoon >= 2.1

Here's some sample code to demonstrate how to upload a file and do the processing of this in an Action.

Example of an upload form:

```
<html>
  <body>
    <form action="upload" method="post" enctype="multipart/form-data">
      File: <input type="file" name="uploadfile" size="50">
      <input type="submit" />
    </form>
  </body>
</html>
```

Example of an Action to get the uploaded file

```
import org.apache.avalon.framework.thread.ThreadSafe;
import org.apache.avalon.framework.parameters.Parameters;
import org.apache.avalon.framework.logger.AbstractLogEnabled;
import org.apache.cocoon.acting.Action;
import org.apache.cocoon.environment.Redirector;
import org.apache.cocoon.environment.SourceResolver;
import org.apache.cocoon.environment.ObjectModelHelper;
import org.apache.cocoon.environment.Request;
import org.apache.cocoon.components.request.multipart.FilePart;
import org.apache.cocoon.components.request.multipart.FilePartFile;

import java.util.Map;
import java.util.Collections;
import java.io.File;

public class UploadTest
  extends AbstractLogEnabled
  implements Action, ThreadSafe
{
    public Map act(Redirector redirector, SourceResolver resolver,
                  Map objectModel, String source, Parameters par)
        throws Exception
    {
        Request request = ObjectModelHelper.getRequest(objectModel);
        FilePart filePart = (FilePart) request.get("uploaded_file");

        File file = ((FilePartFile)filePart).getFile();
        getLogger().debug("Uploaded file = " + file.getCanonicalPath());
        // here you can open an InputStream on the file or whatever
        // you may also want to delete the file after using it
        return Collections.EMPTY_MAP;
    }
}
```

For concept and configuration information for file uploads see [FileUploadsWithCocoon](#)

## Making it work on Cocoon 2.1

If you are using Cocoon >= 2.1, you'll have to change the following imports:

```
import org.apache.cocoon.components.request.multipart.FilePart;
import org.apache.cocoon.components.request.multipart.FilePartFile;
```

to:

```
import org.apache.cocoon.servlet.multipart.Part;
import org.apache.cocoon.servlet.multipart.PartOnDisk;
```

Consequently, the following lines

```
FilePart filePart = (FilePart) request.get("uploaded_file");

File file = ((FilePartFile)filePart).getFile();
```

should read

```
Part filePart = (Part) request.get("uploaded_file");

File file = ((PartOnDisk)filePart).getFile();
```

Warning, this won't work if the servlet container stores the uploaded file in memory (`PartInMemory`) instead of creating a temporary file (`PartOnDisk`).

**Attachment:** [Database](#)