

Contents

- [cfile](#)
- [Syntax](#)
- [Description](#)
- [Body](#)
- [Attributes](#)
- [History](#)

Return to [Tags](#) or [Functions](#)

cfile

Syntax

```
[accept="string"]
[acl="object"]
action="string"
[addnewline="boolean"]
[attributes="string"]
[charset="string"]
[destination="string"]
[file="string"]
[filefield="string"]
[fixnewline="boolean"]
[mode="string"]
[nameconflict="string"]
[output="any"]
[result="string"]
[serverpassword="string"]
[source="string"]
[storeacl="object"]
[variable="string"]>
```

This tag is also supported within cfscrip.

```
file
[accept="string"]
[acl="object"]
action="string"
[addnewline="boolean"]
[attributes="string"]
[charset="string"]
[destination="string"]
[file="string"]
[filefield="string"]
[fixnewline="boolean"]
[mode="string"]
[nameconflict="string"]
[output="any"]
[result="string"]
[serverpassword="string"]
[source="string"]
[storeacl="object"]
[variable="string"];
```

[] = Optional attribute

Description

Handles all interactions with files. The attributes you use with `cfile` depend on the value of the `action` attribute.

For example, if the `action = "write"`, use the attributes associated with writing a text file.

Body

This tag can't have a body.

Attributes

The attributes for this tag are fixed. Except for the following attributes no other attributes are allowed.

Name	Type	Required	Description
accept	string	No	Limits the MIME types to accept. Comma-delimited list. For example, to permit JPG and Microsoft Word file uploads: <code>accept = "image/jpg, application/msword"</code> The browser uses file extension to determine file type.
acl	any	No	An array of struct where each struct represents a permission or grant. Example: <code>[{email="xxx@yyy.com", permission="full_control"}, {group="all", permission="read"}]</code>
action	string	Yes	Type of file manipulation that the tag performs.
addnewline	boolean	No	Yes: appends newline character to text written to file
attributes	string	No	One attribute (Windows) or a comma-delimited list of attributes (other platforms) to set on the file. If omitted, the file's attributes are maintained.
charset	string	No	Character set name for the file contents.
destination	string	No	Absolute pathname of directory or file on web server.
file	string	No	Absolute pathname of file on web server.
filefield	string	No	Name of form field used to select the file.
fixnewline	boolean	No	- true: changes embedded line-ending characters in string variables to operating-system specific line endings - false: (default) do not change embedded line-ending characters in string variables.
mode	string	No	Applies only to Solaris and HP-UX. Permissions. Octal values of UNIX <code>chmod</code> command. Assigned to owner, group, and other, respectively.
nameconflict	string	No	Action to take if filename is the same as that of a file in the directory.
output	any	No	Content of the file to be created.
result	string	No	Name of the result value (default: <code>cffile</code>)
serverpassword	string	No	allow you to access filesystem, also when access is denied for your context
source	string	No	Absolute pathname of file on web server. On Windows, use backward slashes; on UNIX, use forward slashes.
storeacl	any	No	alias for attribute "acl"
variable	string	No	Name of variable to contain contents of text file.

History

Railo 3.3: Recent releases of Railo only allowed you to upload one single at a time if several files were passed to the server within a form. Now Railo supports the action `uploadAll` in the tag which allows a developer to upload all passed files into a certain directory. The tag creates an array of structures containing the information about the uploaded files. Railo 3.3 now allows you as well to upload a file passed by a form within `cfscript` by using the functions `fileUpload()` and `fileUploadAll()`.