

Reference manual



FICO® Xpress Optimization

# FAC File Transfer

Reference Manual

Release 8.3

Last update 7 July, 2015

This material is the confidential, proprietary, and unpublished property of Fair Isaac Corporation. Receipt or possession of this material does not convey rights to divulge, reproduce, use, or allow others to use it without the specific written authorization of Fair Isaac Corporation and use must conform strictly to the license agreement.

The information in this document is subject to change without notice. If you find any problems in this documentation, please report them to us in writing. Neither Fair Isaac Corporation nor its affiliates warrant that this documentation is error-free, nor are there any other warranties with respect to the documentation except as may be provided in the license agreement.

©2014–2017 Fair Isaac Corporation. All rights reserved. Permission to use this software and its documentation is governed by the software license agreement between the licensee and Fair Isaac Corporation (or its affiliate). Portions of the program may contain copyright of various authors and may be licensed under certain third-party licenses identified in the software, documentation, or both.

In no event shall Fair Isaac Corporation or its affiliates be liable to any person for direct, indirect, special, incidental, or consequential damages, including lost profits, arising out of the use of this software and its documentation, even if Fair Isaac Corporation or its affiliates have been advised of the possibility of such damage. The rights and allocation of risk between the licensee and Fair Isaac Corporation (or its affiliates) are governed by the respective identified licenses in the software, documentation, or both.

Fair Isaac Corporation and its affiliates specifically disclaim any warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The software and accompanying documentation, if any, provided hereunder is provided solely to users licensed under the Fair Isaac Software License Agreement. Fair Isaac Corporation and its affiliates have no obligation to provide maintenance, support, updates, enhancements, or modifications except as required to licensed users under the Fair Isaac Software License Agreement.

FICO and Fair Isaac are trademarks or registered trademarks of Fair Isaac Corporation in the United States and may be trademarks or registered trademarks of Fair Isaac Corporation in other countries. Other product and company names herein may be trademarks of their respective owners.

FICO® Xpress Optimization

Deliverable Version: A

Last Revised: 7 July, 2015

Version 8.3

# Contents

---

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Usage examples</b>	<b>2</b>
2.1	File Download example . . . . .	2
2.2	File Upload example . . . . .	2
2.3	File Listing example . . . . .	3
<b>3</b>	<b>Types</b>	<b>4</b>
<b>4</b>	<b>Subroutines</b>	<b>5</b>
	facftdelete . . . . .	6
	facftdownload . . . . .	7
	facftexists . . . . .	8
	facftgetattributes . . . . .	9
	facftlist . . . . .	10
	facftlogin . . . . .	11
	facftlogout . . . . .	12
	facftmkdir . . . . .	13
	facftremovedir . . . . .	14
	facftupload . . . . .	15
	<b>Appendix</b>	<b>16</b>
<b>A</b>	<b>Contacting FICO</b>	<b>16</b>
	Product support . . . . .	16
	Product education . . . . .	16
	Product documentation . . . . .	16
	Sales and maintenance . . . . .	17
	Related services . . . . .	17
	About FICO . . . . .	17
	<b>Index</b>	<b>18</b>

## CHAPTER 1

# Introduction

---

The package and module *facfiletransfer* provides functionality allowing files to be uploaded to and downloaded from the FICO File Transfer Service on the FICO Analytic Cloud. This functionality is available from local models as well as models running in permitted applications on the FICO Analytic Cloud.

Access to the FICO File Transfer Service is authorised by username and password. You must have a username and password in order to use the FICO File Transfer service. The only exception to this is when using a DMP component pre-configured with an FTS username and password.

## CHAPTER 2

# Usage examples

---

## 2.1 File Download example

This example demonstrates download a remote file called `MyFile.txt` from a folder called `MyRemoteFolder` into a local file `MyFile.txt`.

The remote files will be located in a directory that has been provisioned for you on the remote FTS server. This remote directory can be referred to as your root directory `/`, with any subdirectories or files you have created located within this. For example, if you have a file `"MyFile.txt"` in a folder called `"MyFolder"` then you'd refer to this using the remote path `"/MyFolder/MyFile.txt"`.

```
model FacFileTransferExample
uses "facfiletransfer"

parameters
  USER="MyFTSUsername"
  PASS="MyFTSPassword"
end-parameters

declarations
  LOCAL_FILE="MyFile.txt"
  REMOTE_FILE="/MyRemoteFolder/MyFile.txt"
end-declarations

! Login to file transfer service
facftlogin(USER,PASS)

! Download remote file
facftdownload(REMOTE_FILE,LOCAL_FILE)

end-model
```

## 2.2 File Upload example

This example demonstrates upload from a text object in the model to the remote path `/MyRemoteFolder/MyFile.txt`.

```
model FacFileTransferExample
uses "facfiletransfer","mmsystem"

parameters
  USER="MyFTSUsername"
  PASS="MyFTSPassword"
end-parameters

public declarations
```

```
    localData: text
end-declarations

declarations
  REMOTE_FILE="/MyRemoteFolder/MyFile.txt"
end-declarations

! Populate data to upload
localData := "Hello World"

! Login to file transfer service
facftlogin(USER,PASS)

! Download remote file
facftupload(REMOTE_FILE,"mmsystem.text:localData")

end-model
```

## 2.3 File Listing example

This example demonstrates listing all the files in the remote folder `/MyRemoteFolder`.

```
model FacFileTransferExample
uses "facfiletransfer","mmsystem"

parameters
  USER="MyFTSUsername"
  PASS="MyFTSPassword"
end-parameters

declarations
  remoteFiles: list of facftfile
  REMOTE_FOLDER="/MyRemoteFolder"
end-declarations

! Login to file transfer service
facftlogin(USER,PASS)

! Request remote directory listing
remoteFiles := facftlist(REMOTE_FOLDER)

! Output remote filenames
forall( f in remoteFiles) do
  writeln(REMOTE_FOLDER,"/",f.filename)
end-do

end-model
```

## CHAPTER 3

# Types

---

### **factfile** : record

a record containing information about a remote folder or file.

**filename** : string

Filename of the remote file or folder (leafname, not full path)

**isdir** : boolean

true for folders, false for files

**size** : integer

Size of the file, in bytes, or -1 if too large to represent as a Mosel integer

**owner** : string

File owner, given as your numeric user ID in the remote FTS service

**permissions** : string

File permissions, represented as a Unix-style permissions string. This is a three-digit number in which the first digit is the permission for the file's owner, the second digit permissions for the file's group, and the third digit permissions for all other FTS users. For each digit, a value of 0 means no permission, 2 means read-only permission, and 6 means read-write permission.

### **Note**

Modifying the fields within the record will not modify the information on the server.

## CHAPTER 4

# Subroutines

---

<code>facftdelete</code>	Delete a remote file	p. 6
<code>facftdownload</code>	Download a file from the remote path to a local file	p. 7
<code>facftexists</code>	Check whether a remote folder or file exists.	p. 8
<code>facftgetattributes</code>	Query meta-data about a remote file or folder.	p. 9
<code>facftlist</code>	List the files and subfolders within a remote folder.	p. 10
<code>facftlogin</code>	Login to the FAC File Transfer service.	p. 11
<code>facftlogout</code>	Log out of the FAC File Transfer service.	p. 12
<code>facftmkdir</code>	Create a new remote folder.	p. 13
<code>facftremovedir</code>	Delete an empty, remote folder.	p. 14
<code>facftupload</code>	Upload a file to a remote path from a local file	p. 15



## facftdelete

---

### Purpose

Delete a remote file

### Synopsis

```
procedure facftdelete(remote_path:string)
```

### Argument

remote\_path     The path of the remote file to delete, e.g. "/MyFolder/MyFile.txt"

### Example

```
facftdelete("/MyFolder/MyFile.txt")
```

### Further information

1. If the remote file does not exist, the package will raise an IO error.
2. When used from a DMP component with preconfigured FAC File Transfer Service credentials, this function will automatically try to login to the FAC File Transfer Service if you are not already logged in.

## facftdownload

---

### Purpose

Download a file from the remote path to a local file

### Synopsis

```
procedure facftdownload(remote_path:string, local_file:string)
```

### Arguments

<code>remote_path</code>	The path of the remote file to download, e.g. <code>"/MyFolder/MyFile.txt"</code>
<code>local_file</code>	The filename of the local file to which to write the downloaded data.

### Example

```
facftdownload("/MyFolder/MyFile.txt", "MyFile.txt")
```

### Further information

1. The local filename can reference an IO driver in the usual Mosel fashion, e.g. `"mmsystem.text:MyFileContent"`
2. If the remote file does not exist, the package will raise an IO error.
3. If a network error occurs during file download, the package will raise an IO error. Note that in this case, an incomplete file data may have been written to `local_file`.
4. When used from a DMP component with preconfigured FAC File Transfer Service credentials, this function will automatically try to login to the FAC File Transfer Service if you are not already logged in.

## facftexists

---

### Purpose

Check whether a remote folder or file exists.

### Synopsis

```
function facftexists(remote_path:string):boolean
```

### Argument

remote\_path     The path of the remote folder or file, e.g. "/MyFolder/MyFile.txt"

### Return value

true if the file or folder exists on the remote server, false otherwise.

### Example

```
if facftexists("/MyFolder/MySubFolder") then
  writeln("Already exists")
else
  writeln("Does not exist")
end-if
```

### Further information

When used from a DMP component with preconfigured FAC File Transfer Service credentials, this function will automatically try to login to the FAC File Transfer Service if you are not already logged in.

## facftgetattributes

---

### Purpose

Query meta-data about a remote file or folder.

### Synopsis

```
function facftgetattributes(remote_path:string):facftfile
```

### Argument

`remote_path`    The path of the remote file or folder, e.g. `"/MyFolder/MyFile.txt"`

### Example

```
f := facftgetattributes("/MyFolder/MyFile.txt")
writeln(f.file_name," has size ",f.file_size)
```

### Further information

1. The returned size of the file may have been rounded.
2. The file permissions are given in standard Unix format, e.g. 0644 for owner-writable, group-readable, world-readable.
3. If the remote file or folder does not exist, the package will raise an IO error.
4. When used from a DMP component with preconfigured FAC File Transfer Service credentials, this function will automatically try to login to the FAC File Transfer Service if you are not already logged in.

## facftlist

---

### Purpose

List the files and subfolders within a remote folder.

### Synopsis

```
function facftlist(remote_path:string):list of facftfile
```

### Argument

`remote_path`     The path of the remote folder, e.g. `"/MyFolder/MySubFolder"`

### Return value

List containing meta-data for folders and files in the given folder, ordered alphabetically.

### Example

```
declarations
  files: list of facftfile
end-declarations
files := facftlist("/MyFolder/MySubFolder")
forall (f in files) do
  writeln(f.file_name)
done
```

### Further information

1. The returned size of the file may have been rounded.
2. The file permissions are given in standard Unix format, e.g. 0644 for owner-writable, group-readable, world-readable
3. The relative subfolder paths `."` and `.."` are not included in the returned list.
4. This function returns the direct children of `remote_path` only; it is not recursive.
5. If the remote folder does not exist, this function will generate an IO error.
6. When used from a DMP component with preconfigured FAC File Transfer Service credentials, this function will automatically try to login to the FAC File Transfer Service if you are not already logged in.

## facftlogin

---

### Purpose

Login to the FAC File Transfer service.

### Synopsis

```
procedure facftlogin
procedure facftlogin(username:string, password:string)
procedure facftlogin(baseurl:string, username:string, password:string)
```

### Arguments

username	Username for login
password	Password for login
baseurl	URL of the FTS service to use, e.g. <code>https://fts.chi.ficoanalyticcloud.com/api/v1.0</code>

### Example

```
facftlogin('MyUsername', 'MyPassword')
facftlogin('https://fts.chi.ficoanalyticcloud.com/api/v1.0', 'MyUsername', 'MyPassword')
```

### Further information

1. When using the *facfiletransfer* package from your local machine, you are required to login with a username and password prior to making any other calls. When using *facfiletransfer* within a DMP component, login is not required if the component has been preconfigured with a FAC File Transfer username and password.
2. If your username or password are rejected by the FAC File Transfer service, or the FAC File Transfer service is unavailable, the function will return an error and your model will terminate.
3. When used without a username and password, the function will use the credentials preconfigured in your DMP component. This is not necessary for use in the FAC, as the module will login on your first *facfiletransfer* function call, but is provided to allow you to validate credentials at a predictable place.
4. Your username and password are transmitted to the remote FTS server using HTTP Basic authentication across an HTTPS session.
5. When using the *facfiletransfer* package from your local machine, you are required to login with a username and password prior to making any other calls. When using *facfiletransfer* within a DMP component, login is not required if the component has been preconfigured with a FAC File Transfer username and password.
6. If your username or password are rejected by the FAC File Transfer service, or the specified FAC File Transfer service is unavailable, the function will return an error and your model will terminate.
7. When used without a username and password, the function will use the credentials preconfigured in your DMP component. This is not necessary for use in the FAC, as the module will login on your first *facfiletransfer* function call, but is provided to allow you to validate credentials at a predictable place.
8. Your username and password are transmitted to the remote FTS server using HTTP Basic authentication across an HTTPS session.

## facftlogout

---

### Purpose

Log out of the FAC File Transfer service.

### Synopsis

```
procedure facftlogout
```

### Example

```
facftlogout
```

### Further information

Logout is not required (you will be logged out automatically when the model ends). This procedure allows you to end your session earlier for housekeeping reasons.

## facftmkdir

---

### Purpose

Create a new remote folder.

### Synopsis

```
procedure facftmkdir(remote_path:string)
```

### Argument

`remote_path`     The path of the remote folder to create, relative to your root folder on the FTS server, e.g. `"/MyFolder/MySubFolder"`

### Example

```
facftmkdir("/MyFolder/MySubFolder")
```

### Further information

1. If the remote folder already exists, this function will generate an IO error.
2. When used from a DMP component with preconfigured FAC File Transfer Service credentials, this function will automatically try to login to the FAC File Transfer Service if you are not already logged in.



## facftremovedir

---

### Purpose

Delete an empty, remote folder.

### Synopsis

```
procedure facftremovedir(remote_path:string)
```

### Argument

remote\_path     The path of the remote folder to delete, e.g. "/MyFolder/MySubFolder"

### Example

```
facftremovedir("/MyFolder/MySubFolder")
```

### Further information

1. The remote folder must not contain any files or subfolders.
2. When used from a DMP component with preconfigured FAC File Transfer Service credentials, this function will automatically try to login to the FAC File Transfer Service if you are not already logged in.

## facftupload

---

### Purpose

Upload a file to a remote path from a local file

### Synopsis

```
procedure facftupload(remote_path:string, local_file:string)
```

### Arguments

<code>remote_path</code>	The path of the remote file to upload to, e.g. <code>"/MyFolder/MyFile.txt"</code>
<code>local_file</code>	The filename of the local file to which to read the data to upload.

### Example

```
facftupload("/MyFolder/MyFile.txt", "MyFile.txt")
```

### Further information

1. The local filename can reference an IO driver in the usual Mosel fashion, e.g. `"mmsystem.text:MyFileContent"`
2. If the remote file does not exist, it will be created so long as the parent remote folder exists.
3. If the remote file already exists, the previous content will be overwritten.
4. When used from a DMP component with preconfigured FAC File Transfer Service credentials, this function will automatically try to login to the FAC File Transfer Service if you are not already logged in.

## APPENDIX A

# Contacting FICO

---

FICO provides clients with support and services for all our products. Refer to the following sections for more information.

### Product support

FICO offers technical support and services ranging from self-help tools to direct assistance with a FICO technical support engineer. Support is available to all clients who have purchased a FICO product and have an active support or maintenance contract. You can find support contact information on the Product Support home page ([www.fico.com/support](http://www.fico.com/support)).

On the Product Support home page, you can also register for credentials to log on to FICO Online Support, our web-based support tool to access Product Support 24x7 from anywhere in the world. Using FICO Online Support, you can enter cases online, track them through resolution, find articles in the FICO Knowledge Base, and query known issues.

Please include 'Xpress' in the subject line of your support queries.

### Product education

FICO Product Education is the principal provider of product training for our clients and partners. Product Education offers instructor-led classroom courses, web-based training, seminars, and training tools for both new user enablement and ongoing performance support. For additional information, visit the Product Education homepage at [www.fico.com/en/product-training](http://www.fico.com/en/product-training) or email [producteducation@fico.com](mailto:producteducation@fico.com).

### Product documentation

FICO continually looks for new ways to improve and enhance the value of the products and services we provide. If you have comments or suggestions regarding how we can improve this documentation, let us know by sending your suggestions to [techpubs@fico.com](mailto:techpubs@fico.com).

## Sales and maintenance

*USA, CANADA AND ALL AMERICAS*

*Email: XpressSalesUS@fico.com*

*WORLDWIDE*

*Email: XpressSalesUK@fico.com*

*Tel: +44 207 940 8718*

*Fax: +44 870 420 3601*

Xpress Optimization, FICO

FICO House

International Square

Starley Way

Birmingham B37 7GN

UK

## Related services

**Strategy Consulting:** Included in your contract with FICO may be a specified amount of consulting time to assist you in using FICO Optimization Modeler to meet your business needs. Additional consulting time can be arranged by contract.

**Conferences and Seminars:** FICO offers conferences and seminars on our products and services. For announcements concerning these events, go to [www.fico.com](http://www.fico.com) or contact your FICO account representative.

## About FICO

FICO (NYSE:FICO) delivers superior predictive analytics solutions that drive smarter decisions. The company's groundbreaking use of mathematics to predict consumer behavior has transformed entire industries and revolutionized the way risk is managed and products are marketed. FICO's innovative solutions include the FICO® Score—the standard measure of consumer credit risk in the United States—along with industry-leading solutions for managing credit accounts, identifying and minimizing the impact of fraud, and customizing consumer offers with pinpoint accuracy. Most of the world's top banks, as well as leading insurers, retailers, pharmaceutical companies, and government agencies, rely on FICO solutions to accelerate growth, control risk, boost profits, and meet regulatory and competitive demands. FICO also helps millions of individuals manage their personal credit health through [www.myfico.com](http://www.myfico.com). Learn more at [www.fico.com](http://www.fico.com). FICO: Make every decision count™.

# Index

---

## F

- facftdelete, 6
- facftdownload, 7
- facftexists, 8
- facftfile, 4
- facftgetattributes, 9
- facftlist, 10
- facftlogin, 11
- facftlogout, 12
- facftmkdir, 13
- facftremovedir, 14
- facftupload, 15

## L

- login, 11
- logout, 12

## R

- remote file
  - attributes, 9
  - check existence, 8
  - delete, 6
  - download, 7
  - upload, 15
- remote folder
  - attributes, 9
  - check existence, 8
  - create new, 13
  - delete, 14
  - list contents, 10